



Universidad
Carlos III de Madrid

TRABAJO FIN DE GRADO

Herramienta de Análisis de Legibilidad de Contenidos Educativos

Autor: Henar Mata San Juan

Tutor: David Palomar Delgado

Grado: Doble Grado Ingeniería Informática Y ADE

Colmenarejo, septiembre de 2016

Título: HERRAMIENTA DE ANÁLISIS DE LEGIBILIDAD DE CONTENIDOS
EDUCATIVOS

Autor: Henar Mata San Juan

Director: David Palomar Delgado

EL TRIBUNAL

Presidente:

Vocal:

Secretario:

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día ___ de
_____ de 20__ en Colmenarejo, en la Escuela Politécnica Superior de la
Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Resumen

Este proyecto consiste en la investigación sobre los métodos de análisis de legibilidad de contenidos. Sobre todo, se estudia el método Gunning Fog para el análisis de contenido educativo.

Por otro lado, se desarrolla un prototipo de aplicación en la que se podrá insertar un texto y analizar su legibilidad para determinar qué lectores podrán comprender dicho texto más fácilmente.

Finalmente, se mostrará el ciclo de vida que se le puede otorgar al producto añadiendo una serie de acciones que se podrán implementar y desarrollar en un futuro para mejorar su funcionalidad.

Contenido

Contenido.....	4
1. Introducción y objetivos.....	8
1.1. Visión general.....	8
1.2. Motivación	8
1.3. Objetivos	8
1.4. Fases del desarrollo.....	9
2. Estado del arte	9
2.1. Legibilidad de documentos	10
2.1.1. Índices de legibilidad de texto.....	10
2.1.2. Gunning Fog	11
2.2. Tecnologías.....	12
2.2.1. Tecnologías de aplicaciones web	13
2.2.2. Componentes de una aplicación web	14
3. Entorno de desarrollo	14
3.1. Hardware.....	14
3.1.1. Hardware de desarrollo.....	14
3.1.2. Hardware de ejecución	14
3.2. Software	15
3.2.1. Sistema operativo.....	15
3.2.2. Entorno de desarrollo integrado (IDE)	15
3.2.3. Java Development Kit (JDK).....	15
3.2.4. Servidor	15
3.2.5. Librerías.....	16
3.2.6. Aplicaciones de apoyo.....	17
3.3. Instalación del entorno de desarrollo	17
4. Análisis del sistema	18
4.1. Definición del sistema	18
4.2. Establecimiento de requisitos	18
4.3. Casos de uso.....	19
4.3.1. Casos de uso en formato extendido.....	20
7. Diseño del sistema	29
7.1. Arquitectura del sistema	29
7.1.1. Arquitectura física	30
7.1.2. Arquitectura lógica	32
7.2. Diseño estático	33

7.2.1.	Modelo de clases diseño	33
7.2.2.	Diagrama de clases.....	37
7.3.	Diseño dinámico.....	37
7.3.1.	Diagramas de secuencia.....	37
7.4.	Diseño y descripción de la base de datos	43
8.	Pruebas.....	44
8.1.	Diseño de pruebas.....	44
8.2.	Resultado pruebas.....	44
8.2.1.	Registro de usuarios CU-01	45
8.2.2.	Login de usuarios CU-02.....	45
8.2.3.	Salir del sistema CU-03.....	46
8.2.4.	Subir contenido CU-04	46
8.2.5.	Modificar contenido CU-05	47
8.2.6.	Visualizar contenido CU-06	47
8.2.7.	Eliminar contenido CU-07	48
8.2.8.	Crear informe CU-08	48
8.2.9.	Visualizar informe CU-09.....	48
9.	Presupuesto	49
9.1.	Planificación del proyecto y Diagrama de Gantt.....	49
9.2.	Desglose de costes	51
9.2.1.	Horas dedicadas	51
9.2.2.	Coste de personal.....	51
9.2.3.	Coste de material	52
9.2.4.	Coste de software y licencias	53
9.2.5.	Coste de material consumible.....	54
9.2.6.	Coste total del proyecto.....	54
10.	Conclusiones.....	55
10.1.	Conclusiones.....	55
10.2.	Trabajo futuro	56
11.	Bibliografía	58
12.	Anexos.....	60
12.1.	Anexo I: Glosario de acrónimos y definiciones	60
12.1.1.	Acrónimos	60
12.1.2.	Definiciones.....	60
12.2.	Anexo II: Requisitos	61
12.2.1.	Requisitos funcionales.....	61

12.2.2.	Requisitos no funcionales	63
12.3.	Anexo III: Casos de uso	67
12.4.	Anexo IV: Matriz de trazabilidad	69

Índice de ecuaciones

Ecuación 1.	Test de Flesch.....	10
Ecuación 2.	Test de Flesch-Kincaid.....	10
Ecuación 3.	Test de Flesch-Szigriszt	11
Ecuación 4.	Índice de Gunning Fog	11
Ecuación 5.	Fórmula amortización	52

Índice de ilustraciones

Ilustración 1.	Distribución de frecuencias de longitud de palabras.....	12
Ilustración 2.	Modelo aplicación web	13
Ilustración 3.	Arquitectura física	30
Ilustración 4.	Arquitectura lógica	32
Ilustración 5.	Diagrama de clases.....	37
Ilustración 6.	Diagrama de secuencia Registro	38
Ilustración 7.	Diagrama de secuencia Login	39
Ilustración 8.	Diagrama de secuencia Subir contenido	40
Ilustración 9.	Diagrama de secuencia Modificar contenido.....	41
Ilustración 10.	Diagrama de secuencia Visualizar contenido	42
Ilustración 11.	Diagrama de secuencia Eliminar contenido	42
Ilustración 12.	Base de datos	44
Ilustración 13.	Planificación Gantt	50
Ilustración 14.	Caso de uso principal.....	67
Ilustración 15.	Caso de uso autenticación y registro de usuarios.....	67
Ilustración 16.	Caso de uso administrar contenidos	67
Ilustración 17.	Caso de uso administrar informes.....	68
Ilustración 18.	Caso de uso analizar contenido.....	68

Índice de tablas

Tabla 1.	Características ordenador personal.....	14
Tabla 2.	Caso de uso Registro de usuarios	21
Tabla 3.	Caso de uso Login de usuarios.....	23
Tabla 4.	Caso de uso Salir del sistema.....	23
Tabla 5.	Caso de uso Subir contenido	24
Tabla 6.	Caso de uso Modificar contenido	25
Tabla 7.	Caso de uso Visualizar contenido	26
Tabla 8.	Caso de uso Eliminar contenido	27
Tabla 9.	Caso de uso Crear informe	28
Tabla 10.	Caso de uso Visualizar informe.....	29

Tabla 11. Resultado pruebas CU-01	45
Tabla 12. Resultado pruebas CU-02	45
Tabla 13. Resultado pruebas CU-03	46
Tabla 14. Resultado pruebas CU-04	46
Tabla 15. Resultado pruebas CU-05	47
Tabla 16. Resultado pruebas CU-06	47
Tabla 17. Resultado pruebas CU-07	48
Tabla 18. Resultado pruebas CU-08	48
Tabla 19. Resultado pruebas CU-09	48
Tabla 20. Horas dedicadas	51
Tabla 21. Coste personal	52
Tabla 22. Coste material.....	53
Tabla 23. Coste software y licencias	53
Tabla 24. Coste consumibles.....	54
Tabla 25. Coste total	54
Tabla 26. RF-001 Introducir texto	61
Tabla 27. RF-002 Identificar usuario	61
Tabla 28. RF-003 Registrar nuevos usuarios	61
Tabla 29. RF-004 Obtener informe.....	61
Tabla 30. RF-005 Recuperar textos	61
Tabla 31. RF-006 Recuperar informes.....	61
Tabla 32. RF-007 Borrar textos.....	62
Tabla 33. RF-008 Modificar textos	62
Tabla 34. RF-009 Analizar texto.....	62
Tabla 35. RF-010 Cerrar sesión.....	62
Tabla 36. RF-011 Eliminar cuenta.....	62
Tabla 37. RF-012 Eliminar informe	62
Tabla 38. RF-013 Contenido personalizado.....	63
Tabla 39. RF-014 Sesión iniciada	63
Tabla 40. RF-015 Clasificar textos	63
Tabla 41. RS-001 Compatibilidad web	63
Tabla 42. RS-002 Base de datos	63
Tabla 43. RS-003 Tecnología de desarrollo	63
Tabla 44. RS-004 Entorno de desarrollo.....	64
Tabla 45. RSG-001 Sesión usuarios	64
Tabla 46. RSG-002 Contraseñas	64
Tabla 47. RU-001 Feedback usuarios	64
Tabla 48. RU-002 Diferenciación notificaciones	64
Tabla 49. RU-003 Mensajes confirmación.....	65
Tabla 50. RU-004 Lenguaje común.....	65
Tabla 51. RU-005 Guiar usuarios.....	65
Tabla 52. RI-001 Formulario inscripción.....	65
Tabla 53. RI-002 Formulario identificación	65
Tabla 54. RI-003 Formulario subir texto.....	66
Tabla 55. RI-004 Botón eliminar texto	66
Tabla 56. RI-005 Ventana informe	66
Tabla 57. Matriz de trazabilidad.....	69

1. Introducción y objetivos

A lo largo de este apartado se mostrará una visión global del proyecto, así como de las motivaciones y los objetivos del mismo. Con esto, se busca establecer un marco del entorno de desarrollo e investigación del proyecto.

1.1. Visión general

El objeto de este proyecto es realizar una aplicación web que permita los usuarios de la misma obtener el nivel de legibilidad de un texto de contenido educativo a partir de los parámetros establecidos por el índice Gunning Fog.

1.2. Motivación

Con el desarrollo de esta plataforma se busca ofrecer a los usuarios, en especial personal docente, una herramienta que analice los textos que utilizarán para otorgar algún conocimiento a otras personas.

Es común encontrar textos de carácter divulgativo que son difíciles de comprender debido a la poca legibilidad con la que cuentan. Por otro lado, con la evolución de las plataformas de educación online el personal docente busca poder llegar al mayor número de posibles alumnos. Esta última razón obliga a crear contenidos educativos que puedan ser entendidos por personas con diferentes capacidades y con conocimientos muy variados.

Por este motivo, creo que es de gran utilidad una herramienta de acceso web que permita analizar la legibilidad de diferentes contenidos. Además, como la principal orientación es para contenido de educación online lo más conveniente es la creación de una plataforma web.

1.3. Objetivos

El objetivo principal del proyecto es desarrollar una aplicación web que permita a los usuarios analizar la legibilidad de sus contenidos educativos.

Con este proyecto se busca crear una herramienta que los usuarios utilicen de forma continua a la hora de crear contenidos que subirán a las plataformas de educación online que se encuentran en auge.

Además, se les permitirá almacenar sus contenidos e informes anteriores para poder visualizarlos y modificarlos. Por otro lado, se les permitirá organizar los contenidos almacenados en función de los criterios que establezcan por categorías.

1.4. Fases del desarrollo

El proyecto se divide en diferentes fases, cuya duración se puede observar en el diagrama de Gantt de la sección 9.1

- **Fase de estudio:** establece el periodo de tiempo dedicado a estudiar las diferentes herramientas de desarrollo, así como las diferentes formas de estudiar la legibilidad de contenidos.
- **Fase de análisis:** a lo largo de esta fase se establecen los requisitos y casos de uso para conseguir la funcionalidad del sistema.
- **Fase de diseño:** se utiliza para diseñar los diagramas de diseño estático y dinámico del sistema que se explican a lo largo de este documento. Asimismo, se realiza el diseño de la base de datos.
- **Fase de desarrollo:** en esta fase se implementa el código de la interfaz de la aplicación, la lógica y se crea la base de datos.
- **Fase de pruebas:** durante esta fase se diseñan las pruebas y se ejecutan para detectar y solventar errores.
- **Fase de documentación:** esta fase es la utilizada para documentar el resto de fases del proyecto y es la que corresponde a la realización de esta memoria.

2. Estado del arte

En este nuevo apartado se explicará la tecnología sobre la que se va a desarrollar el proyecto.

Ya se ha comentado anteriormente que el objetivo del proyecto es crear una herramienta para estudiar la legibilidad de contenido educativo utilizando como base el Índice Gunning Fog.

Debido a la naturaleza del proyecto se realiza una aplicación web clásica que ofrece todos los servicios necesarios y nos permite centrarnos en el estudio del Índice Gunning Fog.

Para la creación de la aplicación web se utilizarán las tecnologías propias, como son HTML, CSS, Bootstrap, Javascript o las librerías TagLibs.

2.1. Legibilidad de documentos

Según el diccionario de la Real Academia Española la legibilidad es la cualidad de ser leído. Cuanto más fácil de leer y entender sea un texto mayor legibilidad tendrá, por lo que es un factor muy importante a la hora de crear contenido, en especial cuando se trata de contenido educativo [1].

A continuación, se mostrará un conjunto de índices que calcula la legibilidad de contenidos. Todos tienen en cuenta la longitud del texto y de las palabras y frases contenidas en dicho texto. No obstante, algunos se descartarán para el proyecto debido a su orientación al idioma inglés.

Se debe tener en cuenta que para calcular la legibilidad de los contenidos no se estudia la semántica de las palabras, sino su longitud y disposición en el texto.

2.1.1. Índices de legibilidad de texto

Antes de decidir el uso del índice de Gunning Fog como indicador de la legibilidad de un texto se analizaron otros índices disponibles. A continuación, se explicarán los más importantes y por qué se rechazaron para el proyecto.

- **Prueba de facilidad de lectura de Flesch**

Este test evalúa los textos sobre una escala de 100 puntos, teniendo en cuenta el número de palabras por frase y la media de sílabas por palabra. No tiene en cuenta ninguna variable que pueda ser afectada por el idioma en que esté escrito el texto.

La fórmula [2] de este test es:

$$206,835 - 1,01 * \left(\frac{\text{total palabras}}{\text{total frases}} \right) - 84,6 * \left(\frac{\text{total sílabas}}{\text{total palabras}} \right)$$

Ecuación 1. Test de Flesch

- **Prueba de nivel escolar de Flesch-Kincaid**

Este índice evalúa los textos en función del nivel escolar de Estados Unidos. Debido a que buscamos un índice que se adapte al lenguaje español se rechaza desde el principio.

La fórmula [3] de esta prueba es:

$$0,39 * \left(\frac{\text{total palabras}}{\text{total frases}} \right) + 11,8 * \left(\frac{\text{total sílabas}}{\text{total palabras}} \right) - 15,59$$

Ecuación 2. Test de Flesch-Kincaid

- Índice de Legibilidad de Flesch-Szigriszt

Es una adaptación al lenguaje español del índice de Flesch, mencionado anteriormente. La fórmula [4] de este índice es la siguiente:

$$206,835 - 62,3 * \left(\frac{\text{total sílabas}}{\text{total palabras}} \right) - \left(\frac{\text{total palabras}}{\text{total frases}} \right)$$

Ecuación 3. Test de Flesch-Szigriszt

Todos los índices analizados anteriormente tienen en cuenta el total de sílabas por palabras totales del texto. Por esta razón, se decide utilizar el índice de Gunning Fog ya que valora la cantidad de palabras largas que contiene el texto. Creo que es más importante para estudiar la legibilidad tener en cuenta la longitud de las palabras debido a que las palabras cortas conllevan una menor dificultad de comprensión en el lenguaje español.

2.1.2. Gunning Fog

El índice Gunning Fog no es una tecnología en sí mismo. Se trata de un índice que indica la legibilidad de un texto utilizando una serie de características del mismo. Para determinar la legibilidad no se tiene en cuenta la temática del texto, se utilizan parámetros comunes como puede ser el número de sílabas de las palabras que lo conforman. Este índice lo establece una ecuación ideada por Robert Gunning en el año 1952 para identificar la audiencia a la que se puede dirigir un texto determinado.

Este método para establecer la legibilidad de un texto se creó para el idioma inglés y no es igual de exacto para los textos en idioma español debido a que tiene en cuenta la separación de las palabras en sílabas. Para el cálculo del índice de textos en inglés se utiliza la siguiente fórmula:

$$0.4 * \left(\left(\frac{\text{palabras}}{\text{frases}} \right) + 100 * \left(\frac{\text{palabrasdemásde3sílabas}}{\text{palabras}} \right) \right)$$

Ecuación 4. Índice de Gunning Fog

El resultado de esta fórmula nos ofrece la edad mínima que debe tener un lector para que un texto le resulte comprensible.

Aunque en español las palabras son generalmente más largas que en inglés, la frecuencia con la que aparecen este tipo de palabras en textos de ambos idiomas es muy similar. Esto se puede observar en el gráfico siguiente, obtenido del análisis realizado en el documento *Distribución de frecuencias de la longitud de palabras en el español*, [5].

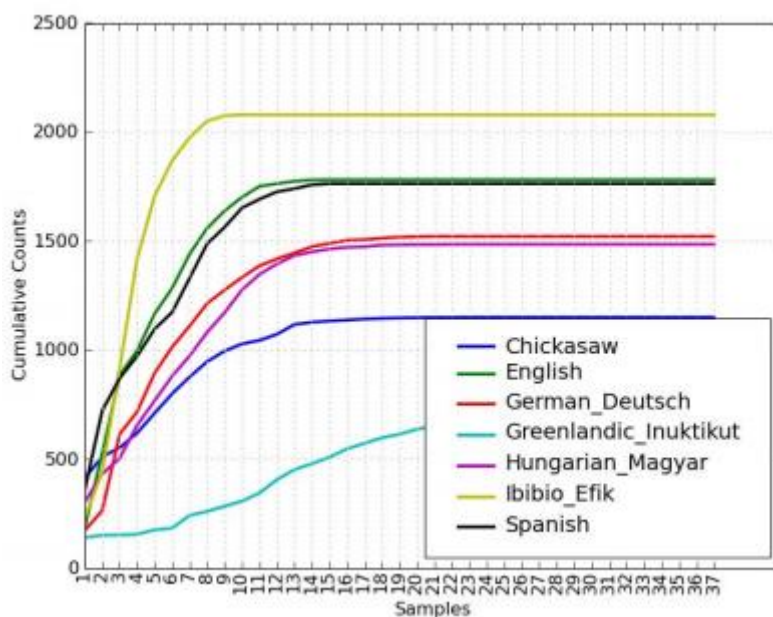


Ilustración 1. Distribución de frecuencias de longitud de palabras

En ese mismo documento, se explica que las palabras de mayor longitud corresponden a palabras científicas, por esta razón creo que esto favorece a que se pueda utilizar la misma fórmula para español que para inglés. Esto se debe a que, tal y como se define el índice de Gunning Fog, no se tiene en cuenta la naturaleza del texto, lo que implica que estudia su dificultad sin tener en cuenta los conocimientos que tenga un lector. Por lo tanto, un texto con muchos conceptos científicos o propios de ciertas áreas de conocimiento serán de difícil comprensión y legibilidad para un alto porcentaje de lectores, es decir, aquellos a los que no se dirija de forma explícita.

A partir del análisis realizado con el documento mencionado anteriormente [5], y otros informes que se citan en la bibliografía se decide utilizar la fórmula creada para el idioma inglés también para el español.

2.2. Tecnologías

Para la implementación de la herramienta se va utilizar el modelo de aplicaciones web clásico.

En este modelo, el usuario hace peticiones a través de un navegador al servidor de aplicaciones. El servidor tras recibir la petición del usuario procesa los datos recibidos y opera con las clases implementadas de la aplicación y la base de datos para dar una respuesta al usuario. El usuario interactúa a través del navegador mediante interfaces que

devuelve el servidor como respuesta a sus peticiones. En este modelo, el servidor y el usuario se quedan esperando hasta que una de las partes realiza o contesta una petición.

La conexión se realiza mediante peticiones http y las interfaces están implementadas con HTML, CSS y Bootstrap. Dentro del servidor se crean clases denominadas servlets que procesan las peticiones de los usuarios y realizan las operaciones necesarias para devolver la respuesta y la interfaz correcta al usuario.

En la siguiente ilustración se puede observar que la interacción del usuario con el servidor es una operación síncrona, ya que debe esperar las respuestas y no se pueden realizar varias consultas al mismo tiempo por el mismo usuario.

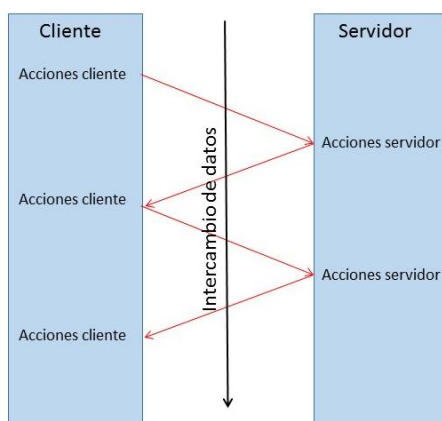


Ilustración 2. Modelo aplicación web

2.2.1. Tecnologías de aplicaciones web

Las aplicaciones web del modelo clásico se componen de diferentes tecnologías que aportan diferentes herramientas al desarrollador para ofrecer la funcionalidad deseada al usuario a la hora de implementar dicha aplicación. Las tecnologías utilizadas para el desarrollo de este proyecto son:

- **HTML5 (HyperText Markup Language) [6]**: es un lenguaje de etiquetas utilizado para estructurar contenidos en el formato estándar utilizado por las páginas web (hipertexto).
- **CSS3 (Cascade Style Sheets) [7]**: en español hoja de estilo en cascada. Es un lenguaje utilizado para dar formato y definir el diseño de contenido estructurado escrito generalmente en lenguaje HTML o XML.

- **Bootstrap [8]:** es un framework que permite hacer los estilos de CSS adaptables a al tamaño del dispositivo desde el que se visualiza de forma sencilla. Fue creado por Twitter que liberó el código en 2011.

2.2.2. Componentes de una aplicación web

- **Servlets:** son clases java que implementan las funcionalidades de la aplicación. Reciben las peticiones y las procesan para devolver a los usuarios la respuesta correcta a su consulta. La conexión a estas clases se realiza mediante http.
- **Interfaces de usuario:** son clases jsp que combinan código HTML+CSS con código Java (Javascript) o, en nuestro caso con el conjunto de librerías Taglibs que permiten optimizar el código y aligerarlo.

3. Entorno de desarrollo

En este apartado del documento se detallará el entorno de desarrollo. Se especificará tanto la parte hardware como la parte software del proceso de desarrollo del proyecto.

3.1. Hardware

3.1.1. Hardware de desarrollo

Se ha optado por utilizar el ordenador personal para el desarrollo de la herramienta de legibilidad. Las características del equipo utilizado son las que se muestran en la tabla siguiente:

Fabricante	Hewlett-Packard
Modelo	HP Envy 15-J100ns
Procesador	Intel® Core™ i7-4702MQ CPU @ 2.20GHz 2.20GHz
Memoria (RAM)	16,0 GB
Tarjeta gráfica	NVIDIA GeForce GT 750M

Tabla 1. Características ordenador personal

3.1.2. Hardware de ejecución

Para la ejecución de la herramienta se necesita un ordenador con conexión a Internet, ya que necesitará conectarse a un servidor para realizar las tareas correspondientes a la herramienta de legibilidad de contenido. No obstante, debido a que el desarrollo se lleva a cabo con un servidor local, se puede utilizar el mismo ordenador que se utiliza para el desarrollo de la aplicación. Por esta razón, se vuelve a usar el ordenador personal que tiene las características definidas en la Tabla 1.

No obstante, para un entorno de producción final será necesario utilizar un servidor conectado a Internet con una IP pública asociada a un dominio. Creemos que lo óptimo sería un servidor Linux con 2Gb de memoria RAM y 50Gb de espacio web.

3.2. Software

3.2.1. Sistema operativo

El sistema operativo del ordenador utilizado para la implementación de la herramienta es Windows 10 Home. No obstante, se utiliza la imagen de un sistema operativo diferente para el desarrollo, haciendo uso de una máquina virtual para la ejecución de la misma.

La imagen del sistema operativo utilizada corresponde a un sistema operativo Ubuntu 14.04 de 64 bits.

3.2.2. Entorno de desarrollo integrado (IDE)

Debido a que la aplicación web será desarrollada en lenguaje Java, se necesitará un IDE adecuado, que en este caso será Eclipse Luna.

La herramienta de desarrollo Eclipse permite desarrollar en lenguaje Java, así como en otros diferentes. Además, permite la inclusión de diferentes plug-ins que permiten la integración de diferentes tecnologías con las que poder trabajar.

Por otro lado, esta herramienta cuenta con una gran documentación sobre la plataforma y es de fácil acceso a través de Internet, lo que facilita la solución de problemas.

La versión utilizada para el desarrollo de la aplicación web es Eclipse Luna.

3.2.3. Java Development Kit (JDK)

La aplicación será desarrollada sobre Java por lo que necesita ser instalado el módulo de desarrollo JDK.

3.2.4. Servidor

Para la correcta implementación y ejecución de la herramienta serán necesarios dos servidores. Primero se necesita un servidor de aplicaciones donde se alojará la herramienta y después un gestor de base de datos para obtener un sistema persistente.

3.2.4.1. Glassfish

Se ha elegido Glassfish 4 como servidor de aplicaciones, que será donde se ejecute la herramienta. Es decir, el servidor será quien reciba las peticiones y realice los cálculos necesarios para devolver al usuario una respuesta a su llamada.

La versión utilizada es Glassfish 4.0, este servidor permite conexión con la base de datos a través de JDBC y JNDI.

Este servidor esta implementado con tecnologías Java EE y permite que sobre él corran aplicaciones web desarrolladas con las mismas tecnologías.

3.2.4.2. *MySQL*

Para el proyecto se utiliza MySQL como gestor de base de datos. Llevará a cabo la ejecución de las sentencias de consulta a la base de datos.

Para la conexión de la base de datos con el servidor de aplicaciones es necesario instalar en este último un driver llamado MySQL Connector.

3.2.5. Librerías

3.2.5.1. *Java Data Base Connectivity (JDBC)*

Se trata de una librería Java que proporciona los medios para la conexión con una base de datos desde una aplicación desarrollada con código Java. Es una librería incluida en el JDK y aporta clases y métodos necesarios para realizar las consultas a la base de datos.

No obstante, para su correcto funcionamiento y que se realice realmente la conexión con la base de datos es necesario instalar el driver MySQL Connector que se ha mencionado anteriormente.

3.2.5.2. *TagLibs*

Son librerías de etiquetas que se pueden integrar en el código de las páginas JSP de proyectos desarrollados con la tecnología J2EE. Con estas librerías se consigue aligerar el código y optimizarlo.

3.2.5.3. *Bootstrap*

Bootstrap es un framework de código abierto que se utiliza para el diseño de sitios web. Ofrece herramientas para el desarrollo de aplicaciones responsivas.

Inicialmente fue creado por Twitter, pero en 2011 liberan el código y en 2012 se convierte en el mayor proyecto de desarrollo de GitHub. En la actualidad, sigue siendo desarrollado conjuntamente por integrantes del equipo de Twitter y de desarrollo de GitHub [9].

Se utiliza una plantilla inicial para el diseño de la aplicación y se modifica a lo largo del desarrollo para adaptar el estilo a la funcionalidad de la aplicación.

3.2.6. Aplicaciones de apoyo

A la hora de realizar el proyecto se han utilizado diferentes herramientas externas al mismo para dar soporte y facilitar su realización.

En primer lugar, se ha utilizado el procesador de textos de Microsoft, es decir, Microsoft Office Word 2016. Esta aplicación se ha utilizado para la redacción de la documentación propia del proyecto, en la que se encuentra esta memoria.

Otra herramienta que se ha utilizado es ArgoUML, se trata de una herramienta para la creación de diferentes clases de diagramas que también se incluyen en este documento. Entre sus usos se encuentran el diagrama de casos de uso, los diagramas de secuencias y los diagramas de clases. Es un software *open source* distribuido bajo la licencia Eclipse Public License (EPL) 1.0, que se puede descargar desde la página web.

3.3. Instalación del entorno de desarrollo

Como se ha comentado anteriormente, se utiliza una imagen de sistema operativo Ubuntu. Para la ejecución de dicha imagen se utiliza la máquina virtual Oracle VM VirtualBox. Por lo tanto, lo primero que se debe instalar es dicho software que se descarga de la página web de VirtualBox [10] y es de licencia GNU.

Tras instalar la máquina virtual y ejecutar la imagen del sistema operativo se debe instalar el entorno de desarrollo de Java, es decir, el software Eclipse y el JDK correspondiente que nos ofrece las herramientas para poder implementar aplicaciones en lenguaje Java. Para instalar Eclipse, se accede a la página web [11] y se descarga el archivo comprimido correspondiente, en este caso es la versión Luna del software.

Para trabajar con la base de datos se utiliza el software libre MySQL Workbench que se puede descargar desde la página web de MySQL [12].

Una vez instaladas las herramientas del entorno de desarrollo se ejecuta Eclipse y se define el servidor que se va a utilizar, en nuestro caso Glassfish 4. Para configurarlo, es necesario descargar el plug-in de Eclipse para el servidor.

Finalmente, después de instalar el servidor se debe configurar el mismo para la conexión con la base de datos a través de la capa de persistencia. Para ello, se accede a la administración del servidor y se configuran las conexiones JDBC con los parámetros correctos.

4. Análisis del sistema

A continuación, se definirá la funcionalidad del sistema para poder describir los requisitos del mismo y establecer los casos de uso pertinentes.

4.1. Definición del sistema

Para definir el sistema debemos decir qué debe hacer para que se considere un funcionamiento correcto en función de las especificaciones establecidas. Por esto, el sistema deberá realizar las siguientes tareas:

- Los usuarios podrán introducir un texto para ser analizado según las características del índice Gunning Fog.
- Ofrecer un informe acerca de los resultados de las pruebas a los usuarios.
- Identificación de usuarios registrados mediante usuario y contraseña.
- Acceso a textos subidos con anterioridad si el usuario está registrado, pudiendo ser eliminados.
- Acceso a informes de textos anteriores si el usuario está registrado.

4.2. Establecimiento de requisitos

Tras definir el sistema, se pueden establecer los requisitos que se deberán cumplir para llegar a satisfacer la funcionalidad del sistema descrito. El conjunto de requisitos se mostrará en el anexo 12.2 .

Para la definición de requisitos se utiliza la especificación IEEE 830. No obstante, se han usado las características de definición que más se adaptan a nuestro proyecto, dado que el estándar IEEE 830 define una larga lista de propiedades que pueden describir un requisito.

Las variables utilizadas son:

- **Identificador:** está compuesto de caracteres que hacen referencia al tipo de requisito y su número. Es único de cada requisito.
- **Nombre:** identifica el requisito de forma única haciendo referencia brevemente a su descripción.
- **Necesidad:** muestra la importancia del requisito para que el sistema cumpla con su funcionalidad. Dispone de tres niveles: **esencial**, el sistema no puede funcionar sin dicho requisito; **deseable**, el sistema mejora su funcionamiento con dicho

requisito; y **opcional**, el sistema es más valioso para el usuario, pero no se modifica en gran parte su funcionalidad.

- **Prioridad:** muestra la necesidad de conseguir un requisito antes que otro, tiene tres niveles: **alta**, **media** y **baja**.
- **Descripción:** da una breve definición de en qué consiste el requisito, de forma clara y concisa para explicar el requisito correctamente y que no queden dudas sobre su funcionalidad dentro del sistema.

4.3. Casos de uso

Tras realizar la descripción de requisitos se definen los casos de uso, estos definen las acciones que los usuarios de la plataforma pueden llevar a cabo.

A continuación, se explicarán los casos de uso obtenidos y cuyos diagramas se muestran en 12.3 Anexo III: Casos de uso.

Para tener una visión global de la funcionalidad del sistema se ha creado un primer diagrama que muestra los casos de uso principales de los que se derivan los siguientes diagramas, este es la imagen **¡Error! No se encuentra el origen de la referencia.** El diagrama de la imagen anterior se desglosa en diagramas más específicos, y son mostrados en las imágenes siguientes del mismo anexo.

El esquema de los diagramas de casos obtenidos es el siguiente:

- Autenticación y registro de usuarios
 1. Registrar usuario
 2. Login usuario
 3. Salir del sistema
- Administrar contenidos
 1. Subir contenido
 2. Modificar contenido
 3. Visualizar contenido
 4. Eliminar contenido
- Administrar informes
 1. Crear informe
 2. Modificar informe
 3. Visualizar informe
 4. Eliminar informe

- Análisis de contenido
 1. Escribir texto
 2. Analizar texto
 3. Crear informe

Con el sistema interactúan dos actores, uno es el usuario no registrado que puede escribir un texto para ser analizado. El segundo actor del diagrama es el usuario registrado que, además de escribir textos para ser analizados, puede guardar sus contenidos, sus informes y crear categorías para organizar los contenidos.

4.3.1. Casos de uso en formato extendido

A lo largo de este apartado se mostrarán las tablas con los casos de uso en formato extendido:

CASO DE USO	Registro de usuarios	
IDENTIFICADOR	CU-01	
ACTOR	Usuario no registrado	
DESCRIPCIÓN	El usuario quiere registrarse para acceder a todas las funcionalidades de la aplicación.	
PRECONDICIÓN	El usuario no debe estar registrado anteriormente	
ESCENARIO NORMAL	ACTOR	SISTEMA
	<ol style="list-style-type: none"> 1. Introduce los datos de registro en el formulario. 2. Envía los datos al servidor de aplicaciones. 	<ol style="list-style-type: none"> 3. Comprueba corrección de los datos introducidos por el usuario. 4. Si los datos son correctos y completos se introducen en la base de datos. 5. Se inicia sesión para el nuevo usuario. 6. Muestra un mensaje de conexión correcta. 7. Muestra la interfaz propia del usuario registrado.
ESCENARIO ALTERNATIVO	ACTOR	SISTEMA
		<ol style="list-style-type: none"> 5. Los datos introducidos no son correctos. 6. Se muestra un mensaje de error para orientar al usuario.
POSTCONDICIÓN	Se añade nuevo usuario a la base de datos y se crea una nueva sesión para el usuario nuevo y tiene acceso a la funcionalidad completa de la plataforma.	

Tabla 2. Caso de uso Registro de usuarios

CASO DE USO	Login de usuarios	
IDENTIFICADOR	CU-02	
ACTOR	Usuario registrado	
DESCRIPCIÓN	El usuario registrado quiere acceder a la plataforma para visualizar sus datos y acceder a todas las funcionalidades de la aplicación.	
PRECONDICIÓN	El usuario debe estar en la base de datos del sistema.	
ESCENARIO NORMAL	ACTOR	SISTEMA
	<ol style="list-style-type: none"> 1. Introduce email y clave en el formulario de acceso. 2. Envía los datos al servidor de aplicaciones 	<ol style="list-style-type: none"> 3. Comprueba corrección de los datos introducidos por el usuario. 4. Comprueba que el usuario está en la base de datos y los datos son correctos. 5. Se inicia sesión para el nuevo usuario. 6. Se recuperan los datos del usuario de la base de datos. 7. Muestra un mensaje de conexión correcta. 8. Muestra la interfaz propia del usuario registrado.
ESCENARIO ALTERNATIVO	ACTOR	SISTEMA
		<ol style="list-style-type: none"> 3. Los datos introducidos no son correctos o no se encuentran en la base de datos.

		4. Se muestra un mensaje de error para orientar al usuario.
POSTCONDICIÓN	Se crea una nueva sesión para el usuario nuevo y tiene acceso a la funcionalidad completa de la plataforma.	

Tabla 3. Caso de uso Login de usuarios

CASO DE USO	Salir del sistema	
IDENTIFICADOR	CU-03	
ACTOR	Usuario registrado	
DESCRIPCIÓN	El usuario identificado quiere salir del sistema	
PRECONDICIÓN	El usuario debe estar autenticado	
ESCENARIO NORMAL	ACTOR	SISTEMA
	1. El usuario pulsa sobre el botón 'salir'	2. Cierra la sesión y muestra la página inicial al usuario sin acceso a sus datos personales.
ESCENARIO ALTERNATIVO	ACTOR	SISTEMA
POSTCONDICIÓN	Se cierra la sesión del usuario, mostrando la interfaz para usuarios no identificados.	

Tabla 4. Caso de uso Salir del sistema

CASO DE USO	Subir contenido	
IDENTIFICADOR	CU-04	
ACTOR	Usuario registrado	
DESCRIPCIÓN	El usuario introduce un texto en el sistema para almacenarlo en su cuenta.	
PRECONDICIÓN	El usuario debe haber iniciado sesión	
ESCENARIO NORMAL	ACTOR	SISTEMA
	<ol style="list-style-type: none"> 1. El usuario añade el texto que desea almacenar en la base de datos y su título. 2. Pulsa el botón “Guardar y Analizar” 	<ol style="list-style-type: none"> 3. Se comprueba que los datos son correctos. 4. Se almacenan los datos en la base de datos. 5. Se muestra la interfaz principal al usuario con un mensaje de operación realizada correctamente.
ESCENARIO ALTERNATIVO	ACTOR	SISTEMA
		<ol style="list-style-type: none"> 4. Los datos no son correctos. 5. Se muestra la interfaz de subida de datos con un mensaje informando del error al usuario
POSTCONDICIÓN	Se muestra la interfaz correcta en función de los resultados.	

Tabla 5. Caso de uso Subir contenido

CASO DE USO	Modificar contenido	
IDENTIFICADOR	CU-05	
ACTOR	Usuario registrado	
DESCRIPCIÓN	El usuario puede modificar el contenido que tiene en la plataforma	
PRECONDICIÓN	El usuario debe estar autenticado	
ESCENARIO NORMAL	ACTOR	SISTEMA
	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón modificar del contenido que desee. 2. Se muestra la interfaz del formulario para cambiar o modificar los datos correspondientes a un contenido. 	<ol style="list-style-type: none"> 3. Se comprueba que el contenido existe en la base de datos. 4. Se recupera el contenido y se modifican los campos seleccionados por el usuario. 5. Se muestra la interfaz principal con un mensaje de actualización correcta.
ESCENARIO ALTERNATIVO	ACTOR	SISTEMA
		<ol style="list-style-type: none"> 2. Se produce un error al modificar la base de datos. 3. Se muestra la interfaz de modificar contenido y un mensaje de error al usuario.
POSTCONDICIÓN	Se actualiza la base de datos y se muestra la interfaz principal.	

Tabla 6. Case de uso Modificar contenido

CASO DE USO	Visualizar contenido	
IDENTIFICADOR	CU-06	
ACTOR	Usuario registrado	
DESCRIPCIÓN	El usuario puede acceder al contenido que tiene en la plataforma	
PRECONDICIÓN	El usuario debe estar autenticado	
ESCENARIO NORMAL	ACTOR	SISTEMA
	1. El usuario selecciona el contenido al que desea acceder.	2. Se comprueba que el contenido existe en la base de datos. 3. Se recupera el contenido y se muestra al usuario en la interfaz principal.
ESCENARIO ALTERNATIVO	ACTOR	SISTEMA
		3. Se produce un error al acceder al contenido de la base de datos. 4. Se muestra la interfaz de selección de contenidos y un mensaje de error al usuario.
POSTCONDICIÓN	Se muestra el texto del contenido.	

Tabla 7. Caso de uso Visualizar contenido

CASO DE USO	Eliminar contenido	
IDENTIFICADOR	CU-07	
ACTOR	Usuario registrado	
DESCRIPCIÓN	El usuario puede eliminar un contenido que tiene en la plataforma	
PRECONDICIÓN	El usuario debe estar autenticado y el contenido debe existir en la base de datos	
ESCENARIO NORMAL	ACTOR	SISTEMA
	1. El usuario selecciona el contenido que desea eliminar.	2. Se comprueba que el contenido existe en la base de datos. 3. Se elimina el contenido de la base de datos y se muestra al usuario el listado de contenidos y un mensaje de eliminación correcta.
ESCENARIO ALTERNATIVO	ACTOR	SISTEMA
		2. Se produce un error al acceder al contenido de la base de datos. 3. Se muestra la interfaz de selección de contenidos y un mensaje de error al usuario.
POSTCONDICIÓN	Se muestra el listado de contenidos no eliminados y un mensaje de eliminación correcta al usuario.	

Tabla 8. Caso de uso Eliminar contenido

CASO DE USO	Crear informe	
IDENTIFICADOR	CU-08	
ACTOR	Usuario registrado y usuario no registrado	
DESCRIPCIÓN	El usuario pulsa el botón analizar para generar el índice de Gunning Fog del texto introducido	
PRECONDICIÓN	El usuario ha introducido un texto.	
ESCENARIO NORMAL	ACTOR	SISTEMA
	1. El usuario introduce un texto en el área de texto y pulsa el botón analizar.	2. El sistema comprueba que los datos introducidos son correctos. 3. Se usa el método implementado para calcular el índice de Gunning Fog del texto introducido. 4. Devuelve al usuario el resultado del análisis. Mostrando la interfaz principal y un mensaje de análisis correcto.
ESCENARIO ALTERNATIVO	ACTOR	SISTEMA
		2. Los datos introducidos no son correctos. 3. Se muestra la interfaz principal y un mensaje de error al usuario.
POSTCONDICIÓN	Se muestra la interfaz principal con mensaje de análisis correcto y una ventana con el resultado del análisis.	

Tabla 9. Caso de uso Crear informe

CASO DE USO	Visualizar informe	
IDENTIFICADOR	CU-9	
ACTOR	Usuario registrado	
DESCRIPCIÓN	El usuario selecciona el contenido del que desea visualizar el análisis.	
PRECONDICIÓN	El usuario ha accedido a sus contenidos para ver los informes.	
ESCENARIO NORMAL	ACTOR	SISTEMA
	1. El usuario pulsa el botón visualizar perteneciente al contenido que desea ver.	2. El sistema comprueba que el contenido y el informe están en la base de datos. 3. El sistema recupera el resultado del análisis. 4. Muestra el resultado y la interfaz principal.
ESCENARIO ALTERNATIVO	ACTOR	SISTEMA
		2. Los datos no se encuentran en la base de datos. 3. Se muestra la interfaz principal y un mensaje de error al usuario.
POSTCONDICIÓN	Se muestra la interfaz principal y una ventana con el resultado del análisis.	

Tabla 10. Caso de uso Visualizar informe

7. Diseño del sistema

Después de analizar el sistema y establecer los requisitos y casos de uso, se procede a explicar el diseño de la plataforma.

7.1. Arquitectura del sistema

La plataforma se desarrollará sobre una arquitectura de tipo cliente-servidor. De este tipo de arquitectura existe una variante que divide las funciones de cada parte de la plataforma

en tres capas, y que será la utilizada para este proyecto. Las capas diferenciadas son las siguientes:

- **Capa de presentación:** es la capa front o de interfaz de usuario. A través de esta capa, los usuarios acceden a la plataforma y se comunican con el sistema.
- **Capa de negocio:** esta capa es la que recibe las peticiones de los clientes y las gestiona, devolviendo la respuesta que corresponda en cada caso. Es la capa intermedia entre la capa de presentación y la de almacenamiento de datos.
- **Capa de almacenamiento de datos:** con esta capa se consigue la persistencia del sistema almacenando los datos de la plataforma. Contiene un gestor de base de datos para interactuar con los datos de la base de datos en función de las peticiones recibidas de la capa de negocio.

7.1.1. Arquitectura física

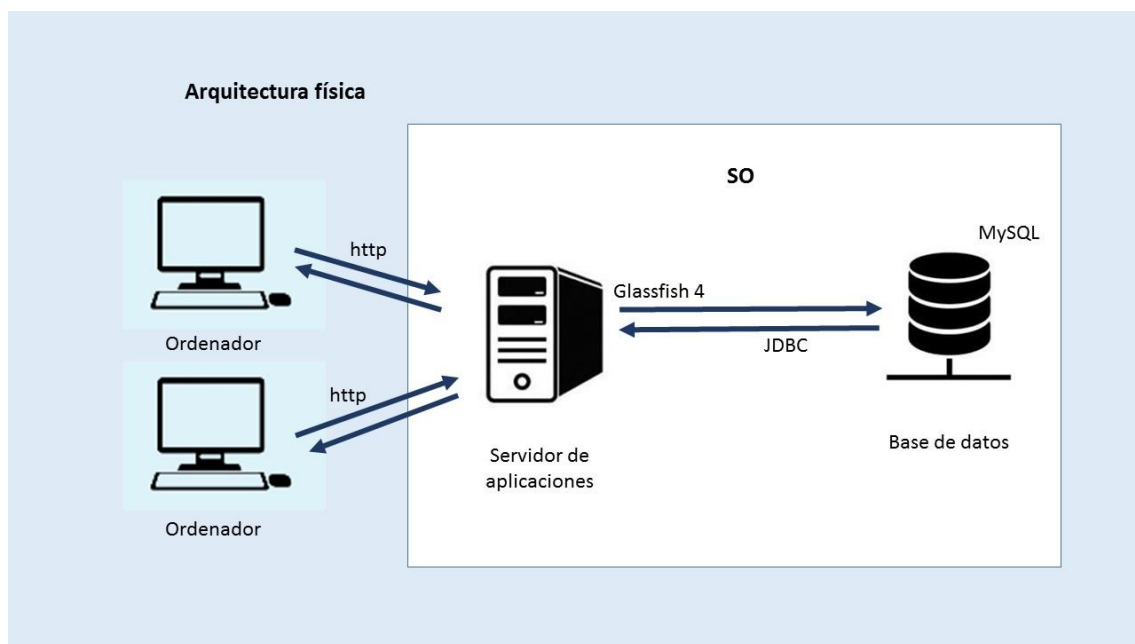


Ilustración 3. Arquitectura física

Respecto a la arquitectura física lo primero que se debe comentar es que la aplicación será alojada en un servidor de aplicaciones Glassfish 4 para su ejecución. Por otro lado, para la gestión de la base de datos utilizada para el almacenamiento de los datos generados por los usuarios se utilizará un sistema gestor de base de datos MySQL.

Aunque se trata de una aplicación web, por cuestiones técnicas se ejecutará en un entorno local (ordenador personal), donde se instalan las herramientas requeridas para el establecimiento del servidor y de la base de datos.

A continuación, se describe el funcionamiento de la aplicación web desarrollada:

Como se ha comentado con anterioridad, la plataforma se alojará en un servidor de aplicaciones Glassfish. Por esta razón, los usuarios accederán a la misma a través de un navegador de Internet, introduciendo la dirección web (URL) de la herramienta en la barra de direcciones.

En la primera página se mostrará el formulario para introducir las credenciales de usuario y un cuadro de texto para analizar los textos introducidos por usuarios no registrados en la plataforma. Tanto si los usuarios se acreditan o si introducen un texto, las tareas se realizarán a través de peticiones http.

En el caso de que se trate de un usuario no registrado que ha introducido un texto, se enviará la petición al servidor de aplicaciones para que realice el análisis y genere el informe sobre la legibilidad del texto.

Por otro lado, si se trata de un usuario registrado, este deberá introducir sus credenciales que serán verificadas contra la base de datos. Como se ha explicado anteriormente, las credenciales serán aportadas por los usuarios a través de peticiones http.

Tras el acceso de un usuario registrado, se accede a la base de datos para recuperar los datos personales de dicho usuario. Después de recuperar los datos se muestra la página personalizada con los documentos del usuario.

Finalmente, el usuario podrá interactuar con la plataforma lo que provoca la conexión con la base de datos para realizar tareas de añadir, eliminar, modificar y consultar los datos personales que tiene almacenado el usuario en el sistema.

Para dar soporte físico a la aplicación se utilizarán los componentes definidos en el apartado 3.2 Software.

7.1.2. Arquitectura lógica

Tras el análisis de la parte física de la arquitectura se pasa a la explicación de la parte lógica de la misma. Para dar soporte a la parte física se utilizarán tres capas lógicas (capa de servicio, capa de dominio y capa de datos) que se detallan a continuación.

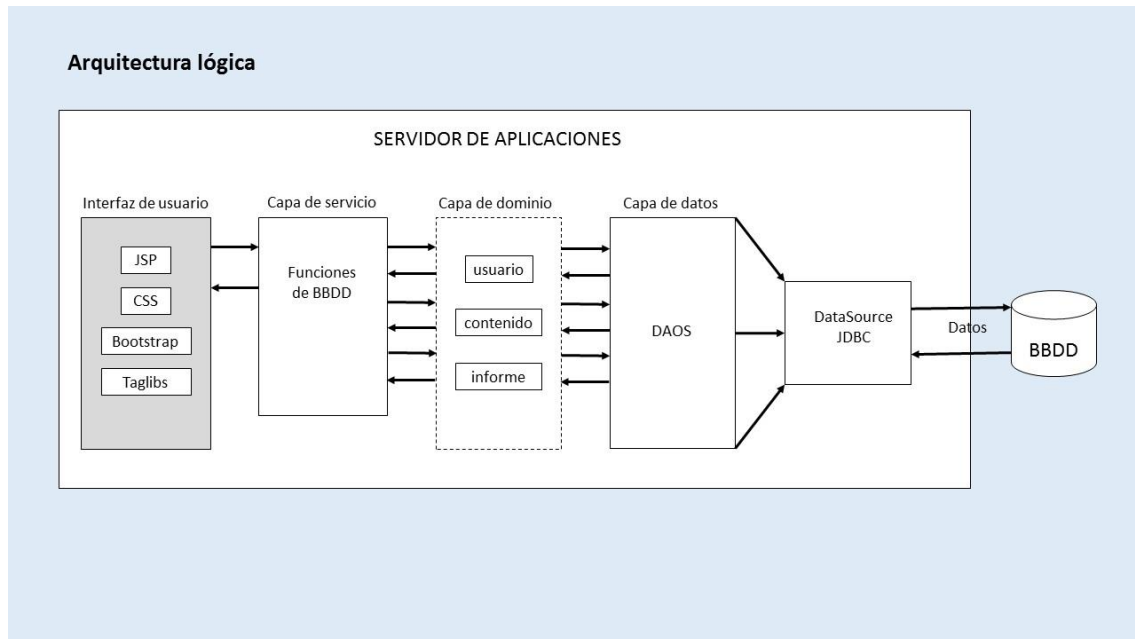


Ilustración 4. Arquitectura lógica

7.1.2.1. Capa de servicio

En esta capa se desarrollarán las funciones que dan funcionalidad al sistema. Cada función será activada a través de las peticiones al servidor que serán realizadas por las acciones llevadas a cabo por los usuarios en la aplicación.

Las peticiones realizadas son asíncronas y se ejecutan mediante Javascript y TagLibs. Sobre estas funciones recae toda la funcionalidad del sistema, lo que incluye el análisis de la legibilidad de los textos, así como los inicios de sesión y todo lo que se muestra de las consultas a la base de datos.

7.1.2.2. Capa de dominio

Esta capa contiene las clases Java que tienen su correspondencia en las tablas de la base de datos de la herramienta. Las clases de esta capa tendrán el mismo nombre y los mismos atributos que la correspondiente en la base de datos. Así, se evita que la capa de servicio comunique a la capa de datos (DAO) parámetros o atributos de forma individual. Puesto que se generan objetos con los atributos de la clase Java de la capa de dominio y se envían a la capa DAO como un objeto completo.

7.1.2.3. *Capa de datos o capa DAO (Data Access Object)*

Esta última capa se utiliza para comunicar la capa de servicio con el gestor de la base de datos utilizado, proporcionando una interfaz de comunicación entre ambos. Debe existir una clase DAO por cada tabla de la base de datos, puesto que cada una realizará las comunicaciones con la base de datos para realizar las consultas provenientes de las peticiones de los clientes. La comunicación de estas clases con la base de datos se realiza a través de JDBC utilizando un DataSource.

La separación de la capa de negocio de la capa DAO aporta una mayor flexibilidad al sistema, así como una independencia que proporciona cierta seguridad. Puesto que la capa de negocio solo recibe los parámetros que provienen de los atributos de la base de datos. De esta forma se evita que la capa de negocio necesite credenciales para acceder a la base de datos.

Esta separación también favorece el desarrollo a la hora y ayuda a resolver errores que se puedan producir de manera más rápida y eficiente.

7.2. *Diseño estático*

7.2.1. *Modelo de clases diseño*

A continuación, se muestran las clases que componen el sistema de la aplicación explicando su funcionalidad, sus atributos y sus métodos. Como se ha explicado en el apartado anterior, existen tres capas que forman la parte lógica de la aplicación, por lo que cada parte tendrá un conjunto de clases Java acorde con su función.

7.2.1.1. *Capa de servicio*

Contiene las clases que albergan los métodos utilizados para desarrollar la funcionalidad de la aplicación. Se separan las clases por la diferente funcionalidad que aportan al sistema.

Los servlets son clases Java que amplían las capacidades del servidor respondiendo a las peticiones http de los clientes. En nuestro caso, ofrecen las operaciones necesarias para responder a las solicitudes de los usuarios a la hora de calcular la legibilidad de un texto o acceder a sus datos almacenados en la base de datos.

Todos los servlets implementados cuentan con los métodos init, doGet, doPost y forward. Dependiendo de las peticiones a las que responda el servlet se accederá al mismo a través del método doPost o doGet.

- **InicioServlet:** este servlet es el responsable de mostrar la interfaz principal al usuario
- **AnalizarServlet:** este servlet recupera los datos introducidos por el usuario en el método doPost y calcula el índice Gunning Fog que genera dicho texto para mostrarle el resultado al usuario. Además de los métodos principales que contienen los servlets, este contiene los métodos calcularPalabras, calcularFrases y calcularPalabrasLargas para poder utilizar los resultados en el cálculo del índice.
- **LoginServlet:** este servlet recupera los datos que introduce el usuario en el formulario de acceso y utiliza la capa DAO para acceder a la base de datos y conectar al usuario con la aplicación.
- **RegistroServlet:** recupera los datos introducidos por el usuario en el formulario de registro, comprueba su validez y utiliza las clases de la capa DAO para almacenar los datos de usuario cuando sean correctos. En caso de ser correctos inicia la sesión del usuario y muestra un mensaje satisfactorio. En caso contrario no almacena los datos en la base de datos y devuelve al usuario a la interfaz de registro con un mensaje de error.

7.2.1.2. *Capa de dominio*

En esta capa se encuentran las clases correspondientes a la base de datos. Esto sirve de intermediario entre la capa de servicio y la de datos facilitando la comunicación entre ambas. Todas las clases contienen como métodos los ‘getters’ y los ‘setters’ de cada atributo.

- **Usuario.java:** esta clase representa a los usuarios registrados en el sistema.
 1. Atributos:
 - nombre: nombre real del usuario, no sirve para identificar al usuario.
 - apellidos: apellidos reales del usuario.
 - email: necesario para registrarse e identificarse en la plataforma. Se usa de identificador de usuario puesto que no existen dos correos electrónicos iguales para dos personas diferentes.
 - password: contraseña del usuario para acceder a la aplicación.
 - id_usuario: número que identifica al usuario de forma unívoca. En este caso no sería necesario debido al email, pero se establece como parámetro para verificar el correcto funcionamiento de la base de datos.

2. Métodos

- ‘*Getters*’ y ‘*Setters*’ de cada atributo.
 - El parámetro `id_usuario` es una variable autogenerada y autoincrementable.
- **Contenido.java:** esta clase representa el contenido que sube el usuario para analizar u legibilidad.
1. Atributos
 - `id_contenido`: identificador del contenido, es único para cada contenido almacenado en la base de datos.
 - `titulo_cont`: título que establece el usuario al contenido.
 - `usuario`: identificador del usuario propietario del contenido.
 - `texto_cont`: texto del contenido.
 - `categoría`: identificador de la categoría a la que pertenece el contenido.
 2. Métodos
 - ‘*Getters*’ y ‘*Setters*’ de cada atributo.
 - El parámetro `id_contenido` es una variable autogenerada y autoincrementable.
- **Informe.java:** esta clase representa el informe generado tras el análisis de legibilidad del contenido.
1. Atributos
 - `id_informe`: identificador del informe, es único para cada informe almacenado en la base de datos.
 - `contenido`: identificador del contenido al que corresponde el informe.
 - `texto_inf`: texto del informe.
 2. Métodos
 - ‘*Getters*’ y ‘*Setters*’ de cada atributo.
 - El parámetro `id_informe` es una variable autogenerada y autoincrementable.

7.2.1.3. *Capa de datos (DAO)*

Las clases de esta capa son interfaces utilizadas para el acceso a los datos de la base de datos. Son las que utilizan las clases de la capa de servicios para acceder a los datos. Los métodos de estas clases son las consultas que se realizarán a la base de datos para insertar,

borrar, modificar o leer datos. Esta capa consta de las interfaces y de las clases que las implementan y contienen el desarrollo de los métodos necesarios para la interacción con la base de datos.

- **UsuarioDao.java**: esta clase se utiliza para implementar las consultas a la tabla usuario realizando la conexión a la base de datos.

1. Atributos:

- EntityManager em
- UserTransaction ut

2. Métodos:

- createUsuario
- modifyUsuario
- deleteUsuario
- findUsuarioByEmail
- findUsuarioByEmailPassword

- **ContenidoDao.java**: esta clase se utiliza para implementar las consultas a la tabla contenido realizando la conexión a la base de datos.

1. Atributos:

- EntityManager em
- UserTransaction ut

2. Métodos:

- createContenido
- modifyContenido

- **InformeDao.java**: esta clase se utiliza para implementar las consultas a la tabla informe realizando la conexión a la base de datos.

1. Atributos:

- EntityManager em
- UserTransaction ut

2. Métodos:

- createInforme
- modifyInforme

7.2.2. Diagrama de clases

Como se ha comentado en el apartado anterior, la aplicación se separa en tres capas (paquetes) que contienen las clases necesarias para implementar la funcionalidad especificada.

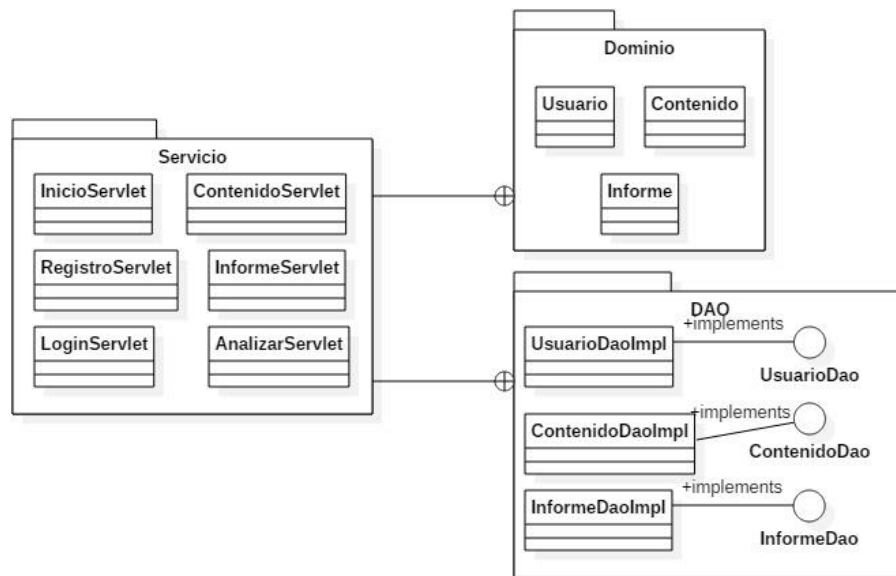


Ilustración 5. Diagrama de clases

7.3. Diseño dinámico

A lo largo de este apartado se mostrarán los diagramas de secuencia correspondientes a las principales funcionalidades de la herramienta. Se desarrolla un diagrama de secuencia por cada caso de uso redactado de la aplicación. Los diagramas de secuencia exponen las acciones de los usuarios, así como el detalle de implementación de la aplicación.

7.3.1. Diagramas de secuencia

Para definir la forma de interactuar de los usuarios con el sistema se utilizan los diagramas de secuencia. Se crea un diagrama de secuencia por cada caso de uso existente. De esta forma se complementa el diseño dando una visualización más completa de la implementación del sistema.

En estos diagramas se muestra la interacción de los objetos, mostrando las clases y los métodos, a través del tiempo. Es decir, contiene los pasos que sigue el usuario o el sistema para realizar cada operación.

7.3.1.1. Registro de usuarios

Para registrar usuarios nuevos en la plataforma, se le muestra al usuario el formulario de registro que debe rellenar y reenviar al servidor de aplicaciones. Primero se comprueba que no existe otro usuario registrado con el mismo email. Después de esta comprobación, se insertan los datos del usuario en la base de datos y se inicia la sesión en el sistema. Se le devuelve al usuario un mensaje de registro y conexión correctos.

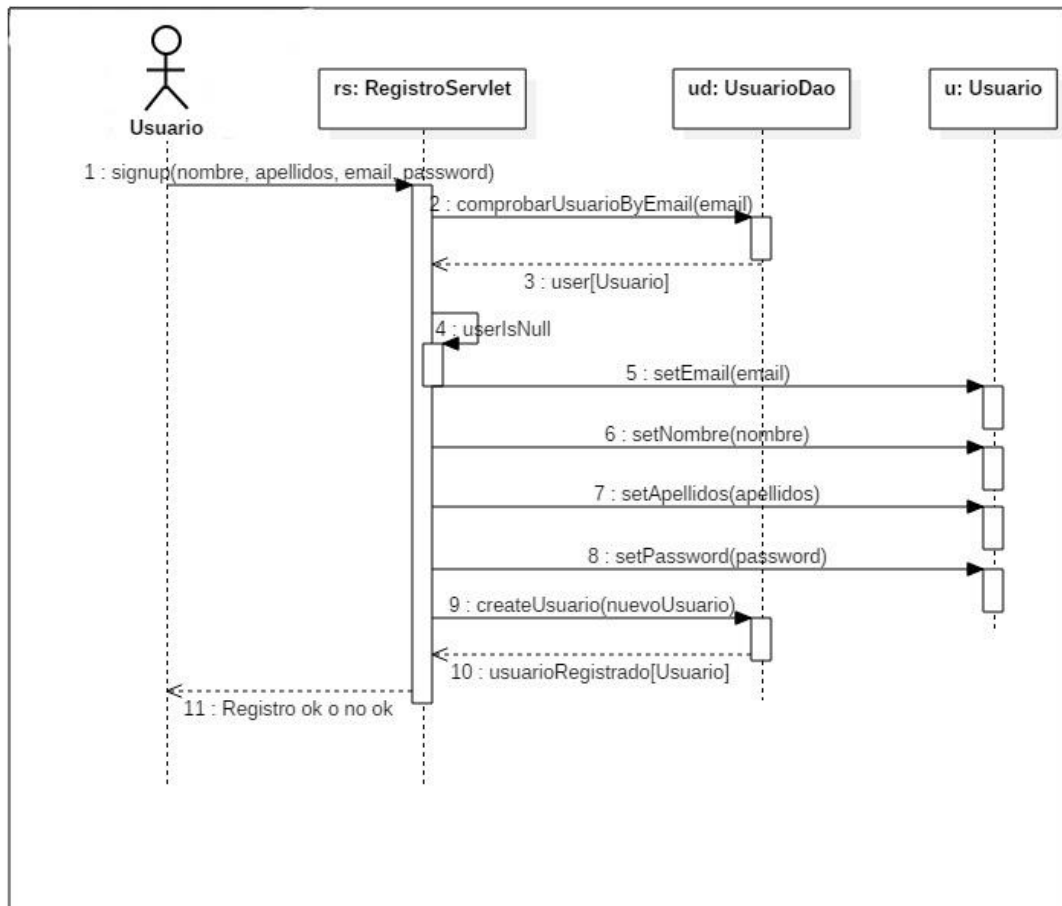


Ilustración 6. Diagrama de secuencia Registro

7.3.1.2. Login de usuarios

Para autenticar a los usuarios lo primero que se hace es comprobar las credenciales que envía el usuario para comprobar que se encuentra registrado en la plataforma. Una vez verificado que se está registrado se realiza su acceso y se le notifica la correcta autenticación en el sistema.

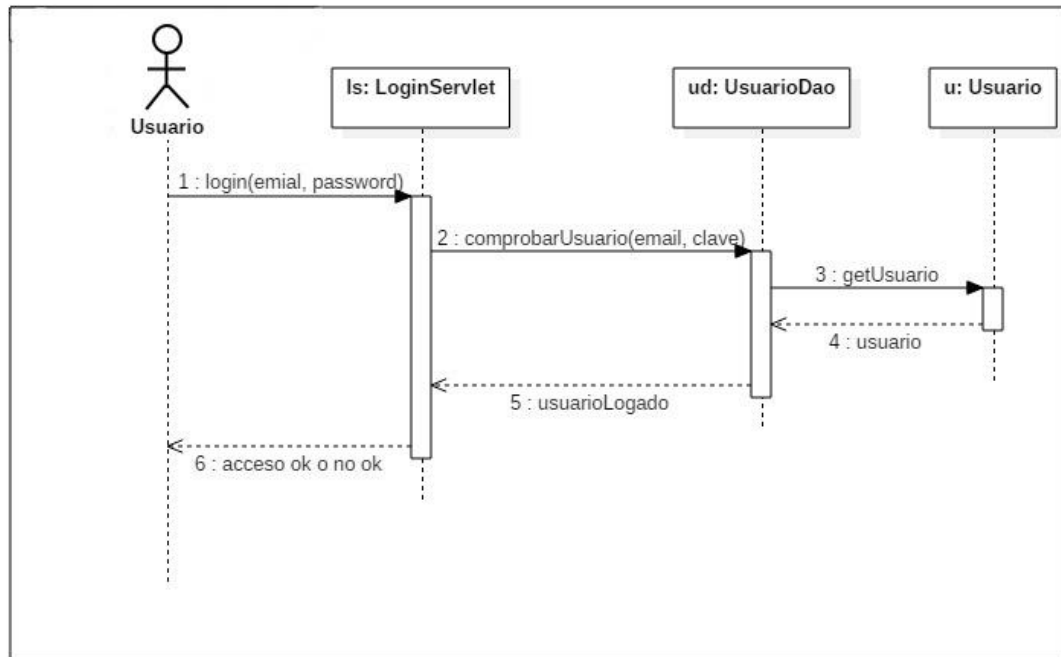


Ilustración 7. Diagrama de secuencia Login

7.3.1.3. Salir del sistema

Debido a la simplicidad del diagrama se omite la imagen. Este diagrama se compone por el usuario del sistema y el servlet LoginServlet. El usuario autenticado solicita cerrar sesión, el servlet comprueba que el usuario se encuentra autenticado, y en caso afirmativo invalida la sesión de dicho usuario.

7.3.1.4. Subir contenido

Tal como se observa en el diagrama siguiente, cuando un usuario añade un contenido a su cuenta, este es analizado previamente. Por esta razón el diagrama de análisis es igual sin la acción de almacenar contenido e informe.

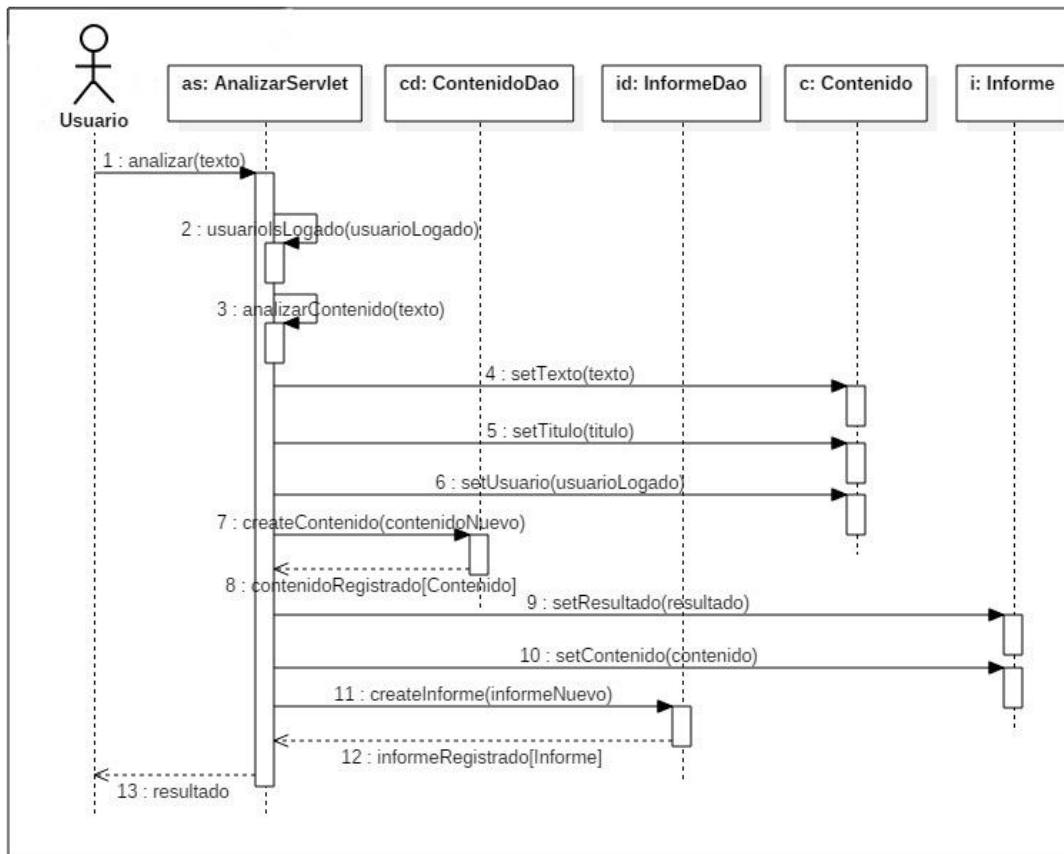


Ilustración 8. Diagrama de secuencia Subir contenido

7.3.1.5. Modificar contenido

El usuario elige un contenido de su lista de contenidos almacenados y solicita modificarlo al servidor. El servidor conecta con el DAO para modificar el texto del contenido. Cuando el contenido es modificado se almacena en la base de datos. Una vez realizada inserción del texto modificado en la base de datos se devuelve un mensaje de modificación realizada al usuario.

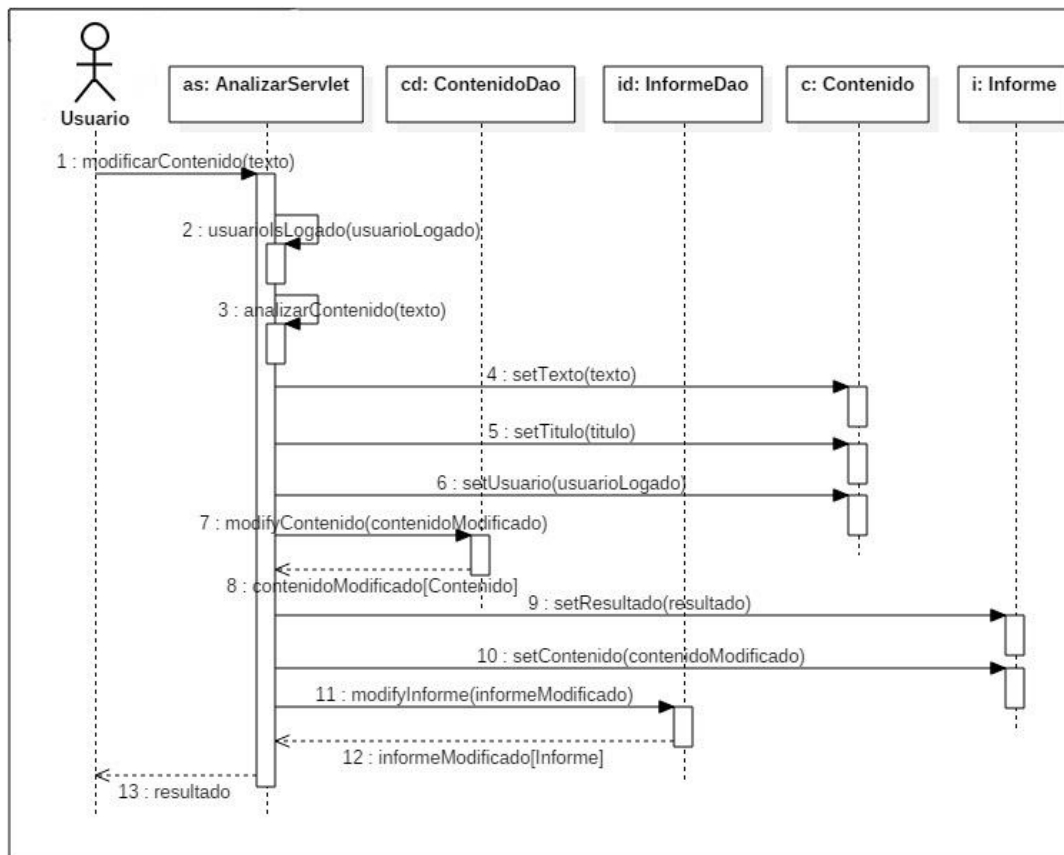


Ilustración 9. Diagrama de secuencia Modificar contenido

7.3.1.6. Visualizar contenido

El usuario selecciona el contenido que desea visualizar y hace la petición al servidor de aplicaciones. El servlet dentro del servidor conecta con la base de datos a través del DAO para recuperar el texto solicitado.

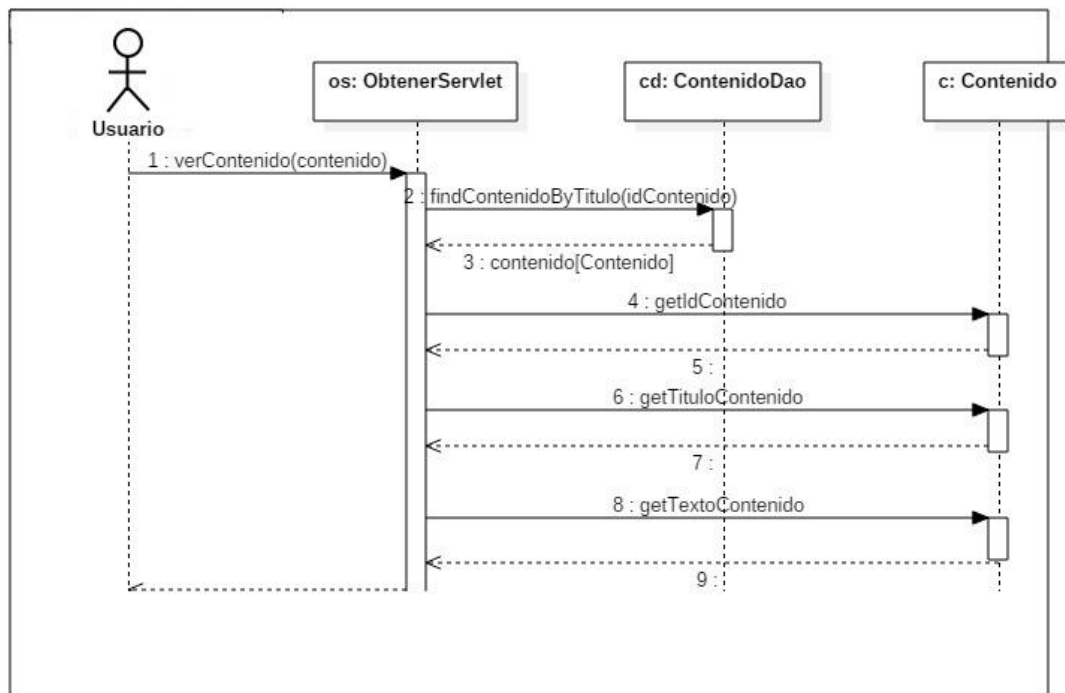


Ilustración 10. Diagrama de secuencia Visualizar contenido

7.3.1.7. Eliminar contenido

El usuario selecciona el contenido que desea eliminar de sus contenidos almacenados y envía la petición al servidor de aplicaciones. Dicho servidor conecta con la base de datos a través del DAO para eliminar el contenido. Desde el servlet del servidor se elimina también el informe relacionado con el contenido que se desea eliminar. Cuando las acciones se han realizado correctamente se devuelve un mensaje al usuario informando de la eliminación del contenido.

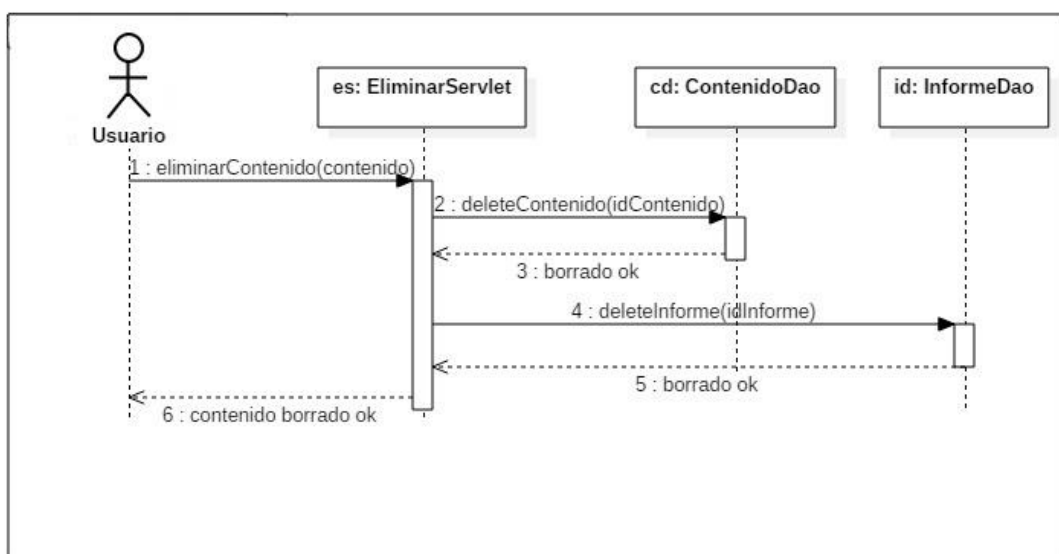


Ilustración 11. Diagrama de secuencia Eliminar contenido

7.3.1.8. *Crear informe*

El diagrama de secuencia para crear un informe se refleja en los diagramas de la Ilustración 8. Diagrama de secuencia Subir contenido e Ilustración 9. Diagrama de secuencia Modificar contenido. Por esta razón, no se muestra la ilustración correspondiente al mismo.

Que el diagrama coincida se debe a que cada vez que se añade o modifica un texto se calcula el índice correspondiente al mismo.

7.3.1.9. *Visualizar informe*

De la misma forma, la imagen del diagrama de visualizar informe no se muestra porque se realizan las mismas actividades que en el diagrama de visualizar un contenido. Sin embargo, las consultas se hacen sobre la tabla de informe de la base de datos en vez de sobre la tabla de contenido.

7.4. *Diseño y descripción de la base de datos*

En este apartado, se mostrará el esquema de la base de datos del sistema. Para esto, es necesario saber que datos recogerá la base de datos y su relación en el sistema.

Sabemos que necesitamos almacenar los datos correspondientes a los usuarios registrados en la plataforma, por lo que contamos con una tabla '**usuario**'. Puesto que los usuarios registrados podrán subir contenido para ser analizado, será necesario almacenar los datos relativos al contenido educativo. Para esto existe la tabla '**contenido**'. El contenido puede ser leído, borrado o modificado por el usuario. La tabla 'contenido' se relaciona con la tabla '**informe**' puesto que cada contenido genera un informe diferente al ser analizado.

Tras conocer las necesidades de almacenamiento del sistema se pasa a establecer que parámetros de cada tabla son necesarios para establecer un sistema persistente y consistente.

En la imagen siguiente se muestra el diagrama correspondiente con la base de datos implementada para nuestro sistema.

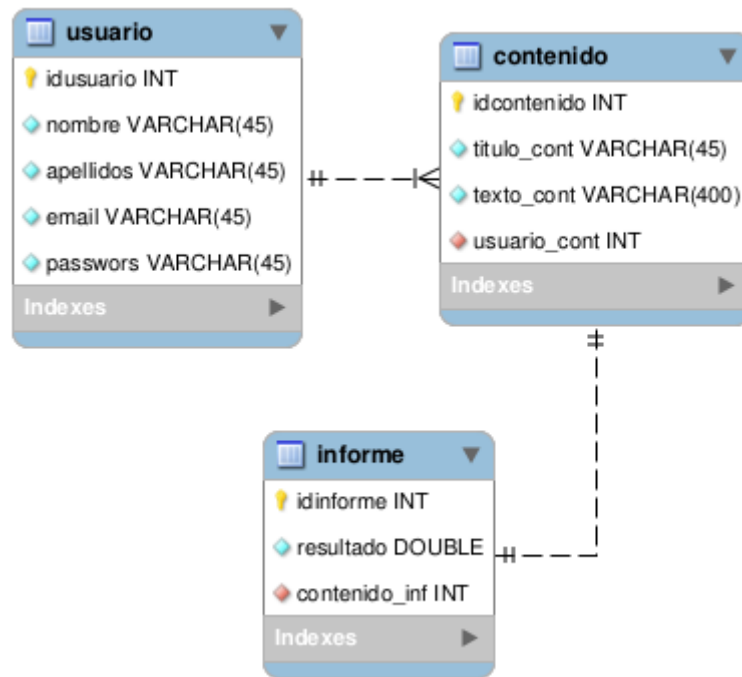


Ilustración 12. Base de datos

8. Pruebas

La última fase de desarrollo del proyecto es la de pruebas. En esta fase se comprobará el correcto funcionamiento de la aplicación, así como el cumplimiento de los requisitos especificados en el análisis del sistema.

Durante la implementación de la aplicación se han ido realizando pruebas de ejecución. pero las realizadas en esta última fase son las que nos ofrecen un resultado fiable. Esto se debe a que ya no se trata de pruebas unitarias, sino que implican la funcionalidad completa de la aplicación.

8.1. Diseño de pruebas

Para poder abarcar todos los posibles fallos de la aplicación y solventarlos, se ha optado por realizar las pruebas en función de los casos de uso desarrollados a partir de los requisitos del sistema.

Por cada caso de uso, se expondrá un escenario de utilización y se comprobará si la respuesta obtenida es la que se esperaba.

8.2. Resultado pruebas

En este apartado se expondrán los resultados obtenidos en las pruebas realizadas sobre cada caso de uso.

8.2.1. Registro de usuarios CU-01

Descripción: un usuario no registrado en la plataforma accede y pulsa el botón 'Registrarse'.				
Nº prueba	Escenario	Entrada	Salida esperada	¿Correcto?
1	Pantalla con formulario de registro	El usuario introduce todos los datos de registro	Interfaz principal con mensaje de registro correcto	Si
2	Pantalla con formulario de registro	El usuario no introduce todos los datos obligatorios	Formulario de registro con mensaje de datos incompletos	Si
3	Pantalla con formulario de registro	El usuario introduce un correo electrónico ya existente en la base de datos	Formulario de registro con mensaje de usuario ya existente	Si
4	Pantalla con formulario de registro	El usuario envía el formulario vacío	Formulario de registro con mensaje de datos incompletos	Si

Tabla 11. Resultado pruebas CU-01

8.2.2. Login de usuarios CU-02

Descripción: un usuario registrado en la plataforma accede y pulsa el botón 'Entrar'.				
Nº prueba	Escenario	Entrada	Salida esperada	¿Correcto?
1	Pantalla principal con formulario de acceso desplegado en el header	El usuario introduce las credenciales de acceso	Interfaz principal con mensaje de acceso correcto	Si
2	Pantalla principal con formulario de acceso desplegado en el header	El usuario no introduce unas credenciales incorrectas	Formulario de registro con mensaje de datos incorrectos	Si
3	Pantalla principal con formulario de acceso desplegado en el header	El usuario no introduce los datos de acceso en el formulario (vacío)	Formulario de registro con mensaje de datos incorrectos	Si

Tabla 12. Resultado pruebas CU-02

8.2.3. Salir del sistema CU-03

Descripción: un usuario registrado y autenticado en el sistema pulsa el botón 'logout' para salir del sistema y cerrar sesión.				
Nº prueba	Escenario	Entrada	Salida esperada	¿Correcto?
1	Página principal con header de usuario conectado	El usuario pulsa el botón 'logout'	Sesión cerrada. Página principal con header sin conexión	Si

Tabla 13. Resultado pruebas CU-03

8.2.4. Subir contenido CU-04

Descripción: un usuario autenticado desea añadir un texto a sus contenidos.				
Nº prueba	Escenario	Entrada	Salida esperada	¿Correcto?
1	Página principal con cuadro de texto para añadir un texto, título y botón para analizar y guardar	El usuario introduce el título y el texto del contenido que desea almacenar y analizar	Interfaz principal con mensaje de almacenamiento correcto y ventana con el resultado del análisis	Si
2	Página principal con cuadro de texto para añadir un texto, título y botón para analizar y guardar	El usuario no introduce todos los datos en el formulario	Página principal de añadir texto con mensaje de aviso de que faltan datos	Si
3	Página principal con cuadro de texto para añadir un texto, título y botón para analizar y guardar	El usuario envía el formulario vacío	Página principal de añadir texto con mensaje de aviso de que faltan datos	Si
4	Página principal con cuadro de texto para añadir un texto, título y botón para analizar y guardar	El usuario introduce un título para el texto que ya existe en la base de datos	Página principal de añadir texto con mensaje de aviso de que el título ya existe	Si

Tabla 14. Resultado pruebas CU-04

8.2.5. Modificar contenido CU-05

<u>Descripción:</u> un usuario autenticado desea modificar un contenido existente				
Nº prueba	Escenario	Entrada	Salida esperada	¿Correcto?
1	Pantalla principal con el texto en el cuadro de texto para analizar	El usuario modifica el texto o el título y pulsa el botón ‘Guardar y Analizar’	Interfaz principal con mensaje de modificación correcta y ventana con el resultado del análisis	Si
2	Pantalla principal con el texto en el cuadro de texto para analizar	El usuario no introduce todos los datos en el formulario	Página principal de añadir texto con mensaje de aviso de que faltan datos	Si
3	Pantalla principal con el texto en el cuadro de texto para analizar	El usuario envía el formulario vacío	Página principal de añadir texto con mensaje de aviso de que faltan datos	Si
4	Pantalla principal con el texto en el cuadro de texto para analizar	El usuario introduce un título para el texto que ya existe en la base de datos	Página principal de añadir texto con mensaje de aviso de que el título ya existe	Si

Tabla 15. Resultado pruebas CU-05

8.2.6. Visualizar contenido CU-06

<u>Descripción:</u> un usuario autenticado pulsa el botón para visualizar un contenido				
Nº prueba	Escenario	Entrada	Salida esperada	¿Correcto?
1	Pantalla con el listado de contenidos del usuario y las funciones que puede realizar sobre el mismo	El usuario pulsa visualizar sobre el botón correspondiente al contenido que desea ver	Pantalla principal con el texto en el cuadro de texto para analizar	Si

Tabla 16. Resultado pruebas CU-06

8.2.7. Eliminar contenido CU-07

<u>Descripción:</u> un usuario autenticado pulsa el botón para eliminar un contenido				
Nº prueba	Escenario	Entrada	Salida esperada	¿Correcto?
1	Pantalla con el listado de contenidos del usuario y las funciones que puede realizar sobre el mismo	El usuario pulsa eliminar sobre el botón correspondiente al contenido que desea borrar	Listado de contenidos del usuario no eliminados y mensaje de eliminación correcta	Si

Tabla 17. Resultado pruebas CU-07

8.2.8. Crear informe CU-08

<u>Descripción:</u> un usuario registrado o no añade un texto para analizar				
Nº prueba	Escenario	Entrada	Salida esperada	¿Correcto?
1	Página principal	El usuario introduce el texto y pulsa analizar	Interfaz principal con mensaje de análisis correcto y ventana con resultado del análisis	Si
2	Página principal	El usuario no introduce texto y pulsa analizar	Página principal con mensaje de aviso de que el texto está vacío	Si
3	Página principal de usuario autenticado	El usuario no introduce texto y pulsa analizar, pero no añade título	Página principal con mensaje de aviso de que faltan datos	Si

Tabla 18. Resultado pruebas CU-08

8.2.9. Visualizar informe CU-09

<u>Descripción:</u> un usuario autenticado desea ver el resultado del análisis de un texto almacenado				
Nº prueba	Escenario	Entrada	Salida esperada	¿Correcto?
1	Página con el listado de contenidos del usuario	El usuario selecciona pulsa el botón de ver informe del texto deseado	Interfaz con el listado de contenidos y ventana con el resultado del análisis del texto	Si

Tabla 19. Resultado pruebas CU-09

9. Presupuesto

9.1. Planificación del proyecto y Diagrama de Gantt

Para la realización del proyecto se utiliza un total de 53 días que se separan en seis fases diferentes. Para mostrar la planificación se utilizará un diagrama de Gantt donde se pueden observar las tareas que componen cada fase, así como su fecha de inicio y de fin. Además, en este diagrama se puede observar si las tareas tienen relación entre sí y a que fase del proyecto pertenecen.

La primera fase que se realiza es el estudio preliminar del problema y los posibles métodos para solventarlo. Además, se estudian las diferentes herramientas de desarrollo y metodologías que se podrán utilizar y se selecciona la más óptima. Por esta razón, en esta fase también se incluye la tarea de instalación del entorno de desarrollo.

La siguiente fase, es la fase de análisis donde se estudia el sistema a implementar y se especifican los requisitos y los casos de uso que definen el sistema y su funcionalidad.

La tercera fase corresponde a la fase de diseño, donde se utilizan los resultados de la fase de análisis para establecer la arquitectura de la aplicación, así como el diseño de la base de datos y los diseños estático y dinámico.

La cuarta fase hace referencia a la fase de desarrollo, en esta fase se definen las principales tareas que se han de implementar en la plataforma para que ofrezca la funcionalidad deseada a los usuarios.

La siguiente fase corresponde a la fase de pruebas, se realizan las pruebas diseñadas para comprobar el correcto funcionamiento del sistema, cumpliendo con los requisitos establecidos. En caso de que sucedan fallos en las pruebas, la solución de los mismos también se contempla en esta fase.

La última fase es la de documentación, esta se desarrolla a lo largo de la duración total del proyecto debido a que en la memoria que se entrega al final contiene la documentación de todas las tareas que se van realizando mientras se va desarrollando el proyecto.

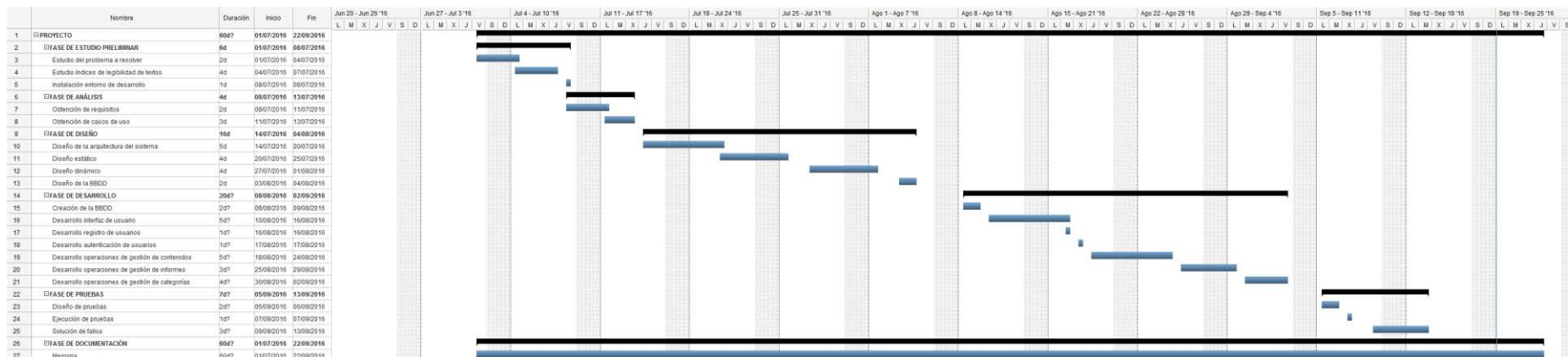


Ilustración 13. Planificación Gantt

Para la realización de este diagrama se ha utilizado la herramienta online de software libre Ganttter, a la que se puede acceder desde la siguiente página web: <https://gantter.com>

9.2. Desglose de costes

Para calcular el coste total del proyecto y el presupuesto que se deberá ofrecer a lo largo de este apartado se explicarán los gastos del proyecto desglosados según su naturaleza.

9.2.1. Horas dedicadas

Lo primero que se debe tener en cuenta para calcular los gastos del proyecto son las horas que se necesitan para su desarrollo. Estas horas se muestran en la siguiente tabla y se obtienen de la planificación mostrada en el apartado anterior.

	DÍAS	HORA/DÍA	HORAS
FASE DE ESTUDIO	6	8	48
FASE DE ANÁLISIS	4	8	32
FASE DE DISEÑO	16	8	128
FASE DE DESARROLLO	20	8	160
FASE DE PRUEBAS	7	4	28
FASE DE DOCUMENTACIÓN	60	4	240
			TOTAL 636

Tabla 20. Horas dedicadas

Según la tabla anterior y los cálculos de planificación del proyecto se ve que el proyecto tiene una duración de 60 días con un total de 636 horas.

9.2.2. Coste de personal

Para realizar cada una de las fases especificadas anteriormente, se requiere de personal cualificado. Por esto, en la siguiente tabla se desglosa el coste de contratación de cada componente del equipo especificando la dedicación mensual de cada trabajador. Los datos que se muestran en la tabla se han obtenido de una estimación a partir de datos reales publicados.

Para calcular la dedicación mensual de los empleados, dividimos el total de horas mensuales de cada empleado (80 horas) entre el número total de horas dedicadas al proyecto dependiendo del puesto que desempeñen.

Tras realizar estos cálculos obtenemos el coste total de personal por función en el proyecto y el coste total del proyecto que asciende a 8.475,24 € sin IVA.

	HORAS	€/HORA	DEDICACIÓN (HOMBRE/MES)	COSTE TOTAL (€)
ANALISTA	32	19,25	0.4	616
DISEÑADOR WEB	128	13,50	1,6	1728
PROGRAMADOR	236	15,79	2,95	3726,44
DOCUMENTACIÓN	240	10,02	3	2404,80
				8.475,24

Tabla 21. Coste personal

9.2.3. Coste de material

A lo largo de este apartado se desglosan los costes ocasionados por el material utilizado en el desarrollo del proyecto. Para calcular la amortización del material se utiliza la siguiente fórmula, obtenida de los esquemas proporcionados por la asignatura de Dirección de Proyectos de Desarrollo Software de la Universidad Carlos III de Madrid.

$$\frac{A}{B} * C * D$$

Ecuación 5. Fórmula amortización

Donde:

- A: número de meses que se utiliza el equipo desde la fecha de facturación
- B: periodo de depreciación (60 meses)
- C: coste del equipo
- D: porcentaje de dedicación al proyecto

De esta forma, y tal como se observa en el resultado de la siguiente tabla, el gasto total del material utilizado tras la amortización es de 43,12 € sin IVA.

	UNIDADES	COSTE (€)	COSTE SIN IVA (€)	DEDICACIÓN (MESES)	COSTE IMPUTABLE (€)
Ordenador HP Envy 15-J100ns	1	1000	826,45	3	41,32
PENDRIVE SanDisk 16 GB	1	8,90	7,35	3	0,37
DISCO DURO EXTERNO SEAGATE 1TB	1	69	57,02	1,5	1,43
					43,12

Tabla 22. Coste material

9.2.4. Coste de software y licencias

La mayoría del software utilizado tanto para el desarrollo como para la documentación posee licencia de software el libre. El software que no tiene licencia de software libre es el paquete de Microsoft Office 2016 utilizado para la redacción de la memoria. La licencia de cuatro años de este producto en la versión Office 365 Universitarios tiene un coste de 65,29 € sin IVA. Por lo tanto, el total del coste de este apartado corresponde al valor de la licencia de Microsoft Office 2016, 65,29 € sin IVA.

No obstante, en la siguiente tabla se puede observar el desglose de software utilizado.

	UNIDADES	COSTE/UNIDAD (€)	COSTE TOTAL (€)
Microsoft Office 2016	1	65,29	65,29
Editor Sublime3	1	0,00	0,00
ArgoUML	1	0,00	0,00
StarUML	1	0,00	0,00
Glassfish Server	1	0,00	0,00
Oracle VM VirtualBox	1	0,00	0,00
MySQL	1	0,00	0,00
Eclipse Luna	1	0,00	0,00
			65,29

Tabla 23. Coste software y licencias

Aunque el entorno de desarrollo no tiene costes de licencias, para el entorno de producción deseable si tendremos un coste de 14,90 €/mes sin IVA por la contratación

del servicio de servidor y dominio. Este precio se obtiene de los servicios que ofrece la empresa arsys [13] para hosting.

9.2.5. Coste de material consumible

Para el desarrollo del proyecto se utilizan diferentes materiales consumibles que se expone a continuación y que conllevan un coste de 4,54€ sin IVA.

	UNIDADES	COSTE/UNIDAD (€)	COSTE TOTAL (€)
Cuaderno	1	3,06	3,06
Bolígrafo	1	0,24	0,24
Lapicero	2	0,32	0,64
Goma de borrar	1	0,66	0,66
			4,54

Tabla 24. Coste consumibles

9.2.6. Coste total del proyecto

A partir de los datos obtenidos en los apartados anteriores, se realiza la suma de todos los costes desglosados para obtener el coste total del proyecto. A los costes anteriormente detallados se deben añadir los costes indirectos del proyecto que se estiman en un 20% obtenido también de la plantilla de la asignatura Dirección de Proyectos de Desarrollo Software de la UC3M.

De esta forma y como se muestra en la siguiente tabla, el coste total del proyecto es de 10.305,83 € sin IVA.

TIPO	COSTE (€)
Coste Personal	8.475,24
Coste material	43,12
Coste software	65,29
Coste consumibles	4,54
Costes indirectos (20%)	1717,64
	10.305,83

Tabla 25. Coste total

10. Conclusiones

A lo largo de este apartado se expondrán las conclusiones a las que se han llegado tras el análisis de los diferentes índices para el estudio de la legibilidad de contenidos. Así como de las decisiones de diseño de la aplicación.

Además, se definirán tareas que se podrían realizar en un futuro para mejorar la aplicación, tanto en la funcionalidad ofrecida como en su ejecución.

10.1. Conclusiones

El objetivo principal de este proyecto era encontrar un método que permitiera analizar la legibilidad de contenidos educativos a los usuarios de una herramienta. Para ello se estudiaron diferentes métodos de análisis y, después se prosiguió a encontrar la mejor forma de ofrecer la herramienta a los usuarios.

Tras el estudio de diferentes métodos de cálculo del nivel de legibilidad de contenido, se decidió utilizar el método de Gunning Fog puesto que era el único índice que se calculaba teniendo en cuenta el porcentaje de palabras largas (más de 3 sílabas) que contenía el texto de estudio.

Además, el cálculo del índice de Gunning Fog da como resultado la edad media de los lectores que comprenderían el texto al leerlo. Mientras que el resto de índices estudiados dan valores entre 0 y 100, donde el mayor valor corresponde a los textos más legibles.

Por otro lado, respecto a la forma de ofrecer la herramienta a los usuarios, nos decantamos por una aplicación web. Esto se debe a que principalmente su uso estará orientado a personal docente que ofrezca sus recursos educativos a través de plataformas educativas online.

De esta forma, facilitamos el uso de nuestra herramienta puesto que ofrece una interfaz fácil de usar. Además, los usuarios podrán utilizar el mismo navegador que utilicen para subir sus contenidos a la plataforma educativa que utilicen.

Se podría implementar una aplicación móvil, pero se decide utilizar el framework Bootstrap para hacer una interfaz adaptable por si algún usuario utiliza un dispositivo móvil para acceder. Esto se debe a que la mayoría de contenidos educativos se crea a través de un ordenador, por lo que lo más lógico es que la herramienta se utilice también, en su mayoría, desde un navegador de un ordenador.

Una vez se decide realizar una aplicación web se decide utilizar el lenguaje Java para la implementación de la funcionalidad. La interfaz de la aplicación se realizará mediante HTML, CSS y el framework Bootstrap.

Finalmente, debido al uso de Java para la codificación se decide utilizar el servidor de aplicaciones Glassfish puesto que está desarrollado con esta tecnología y es de software libre. Además, permite el uso de un conector con el gestor de base de datos de MySQL que es la base de datos que se utiliza.

Como conclusiones personales, puedo decir que gracias a la elaboración de este proyecto he profundizado en el conocimiento de uso y funcionalidad de las herramientas de desarrollo web utilizadas durante la implementación de la plataforma.

Respecto a la parte de código no se puede decir que haya adquirido todo el conocimiento posible debido a las extensas posibilidades que ofrece. No obstante, he adquirido facilidad para generar código, identificar errores en el mismo y depurarlo.

Todos los lenguajes de programación utilizados (Java, HTML5, CSS3), así como Bootstrap ya los había usado durante los años de estudio en la UC3M. Sin embargo, con el desarrollo de este proyecto he conseguido ampliar considerablemente mi conocimiento sobre ellos.

No debo olvidar lo que me ha aportado el estudio sobre los índices de legibilidad. Con ello he descubierto lo importante que es crear un texto que sea entendible por una gran parte de la audiencia. Sobre todo, me ha enseñado que el tema de un texto no influye para que sea cómodo de leer y comprender.

10.2. Trabajo futuro

El proyecto desarrollado podrá ser el inicio de una aplicación web que se podría incluir en diferentes sitios web orientados a la educación online. Actualmente, la ejecución se realiza sobre un entorno local, por lo que el almacenamiento de las credenciales de los usuarios se realiza en la misma base de datos que el resto de datos almacenados. En el futuro sería aconsejable crear un directorio de usuarios para aumentar la seguridad de la plataforma.

En un futuro, teniendo en cuenta la evolución de la tecnología y de los medios que utilizamos para acceder a la misma, se podría realizar el proyecto sobre una plataforma Android o una aplicación AJAX que ofrecerán una mayor interacción con el usuario final.

No obstante, con el uso del framework Bootstrap que se ha mencionado a lo largo del trabajo, no sería tan urgente puesto que ya nos da la posibilidad de integrarlo en dispositivos de diferentes tamaños. Aunque, con la tecnología AJAX se haría una comunicación más ágil entre cliente y servidor.

Por otro lado, se recomendable mejorar la fórmula que calcula el índice de legibilidad para adaptarlo mejor al idioma español con la colaboración de diferentes expertos en filología española.

Otra opción sería acoplar el resto de métodos a la plataforma para que cada usuario elija el método que desea utilizar para analizar sus contenidos, así como un selector de idioma. Si se acoplan diferentes índices se debería introducir un detector de idioma automático para que se realizase el estudio de legibilidad con el índice más favorable para cada idioma.

Respecto a la interfaz del usuario y a las tareas que puede realizar sobre la plataforma, sería recomendable ofrecer a los usuarios la posibilidad de crear categorías para almacenar los diferentes contenidos de su perfil.

Respecto a los resultados, se podrían ofrecer consejos sobre cómo mejorar la legibilidad del texto analizado. Una forma podría ser creando un diccionario de sinónimos donde se convirtieran las palabras largas (mala legibilidad) por palabras cortas (mejor legibilidad).

Finalmente, una mejora de la plataforma sería adaptar la subida de contenidos a diferentes formatos. En la actualidad solo se permite subir texto plano, pero los contenidos educativos raramente se desarrollan en este tipo de texto.

11. Bibliografía

- [1] V. F. Belart, «La legibilidad: un factor fundamental para comprender un texto,» *Atención Primaria*, vol. 34, nº 3, pp. 143-146, Julio 2004.
- [2] X. Ganzábal García, Aplicaciones técnicas de usabilidad y accesibilidad en el entorno cliente, S.A. Ediciones Paraninfo, 2015, p. 39.
- [3] A. König, A. Dengel, K. Hinkelmann, K. Kise, R. J. Howlett y L. C. Jain, Edits., Knowledge-Based Intelligent Information and Engineering Systems, vol. 3, Wellington: Springer, 2004.
- [4] J. Perez Colomé, Cómo escribir claro, S. El Ciervo 96, Ed., Barcelona: Editorial UOC, 2011.
- [5] A. Frías Delgado, «DISTRIBUCIÓN DE FRECUENCIAS DE LA LONGITUD DE LAS PALABRAS EN ESPAÑOL».
- [6] W3C Working Group, «HTML5: W3C,» 28 Octubre 2014. [En línea]. Available: <https://www.w3.org/TR/html5/>. [Último acceso: 2016].
- [7] W3C Working Group, «CSS: W3C,» 13 Octubre 2015. [En línea]. Available: <https://www.w3.org/TR/CSS/>. [Último acceso: 2016].
- [8] M. Otto, J. Thornton, C. Rebert, J. Thilo, H. Fenkart, P. H. Lauke y X. , «Bootstrap,» [En línea]. Available: <http://getbootstrap.com/>.
- [9] M. Otto, J. Thornton, C. Rebert, J. Thilo, H. Fenkart, P. H. Lauke y X. , «About: Bootstrap,» 2016. [En línea]. Available: <http://getbootstrap.com/about/>. [Último acceso: 2016].
- [10] Oracle, «VirtualBox,» 2016. [En línea]. Available: <https://www.virtualbox.org/>. [Último acceso: 2016].
- [11] Eclipse Foundation, «Downloads: Eclipse,» 2016. [En línea]. Available: <https://www.eclipse.org/downloads/>.
- [12] Oracle Corporation and/or its affiliates, «MySQL,» 2016. [En línea]. Available: <http://dev.mysql.com/downloads/workbench/>.
- [13] Arsys, «Hosting: Arsys,» 2016. [En línea]. Available: <https://www.arsys.es/hosting>. [Último acceso: 2016].
- [14] D. Guajardo, «El índice de niebla de Gunning: No me canso de escribir: Blogspot,» 14 Agosto 2012. [En línea]. Available: <http://no-me-canso.blogspot.com.es/2012/08/el-indice-de-niebla-de-gunning.html>. [Último acceso: 2016].
- [15] P. Deapié, «Legibilidad. El índice de Gunning - El misántropo digital: Blogspot,» 10 Octubre 2011. [En línea]. Available: <http://elmisantropodigital.blogspot.com.es/2011/10/legibilidad.html>.
- [16] Wikipedia, «Gunning Fog index: Wikipedia,» 8 Julio 2016. [En línea]. Available: https://en.wikipedia.org/wiki/Gunning_fog_index.

- [17] M. González T. y S. Mattar V., «El índice de niebla y la escritura científica,» *Revista MVZ Córdoba*, vol. 18, nº Supl.1, Diciembre 2013.
- [18] Collabnet, «ArgoUML: Tigris.org,» 2016. [En línea]. Available: <http://argouml.tigris.org/>. [Último acceso: 2016].
- [19] Eclipse, «Eclipse Public License - v 1.0: Eclipse,» [En línea]. Available: <http://www.eclipse.org/legal/epl-v10.html>. [Último acceso: 2016].
- [20] wikiHow, «Programación: wikiHow,» [En línea]. Available: <http://es.wikihow.com/dividir-cadenas-en-Java>. [Último acceso: 2016].
- [21] Real Academia Española, «Real Academia Española,» 2016. [En línea]. Available: <http://www.rae.es/>. [Último acceso: 2016].

12. Anexos

12.1. Anexo I: Glosario de acrónimos y definiciones

12.1.1. Acrónimos

DAO	Data Access Object
GPL	Licencia General Pública de GNU
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
	Integrated Development Enviroment (Entorno de Desarrollo Integrado)
IDE	
IVA	Impuesto sobre el Valor Añadido
J2EE	Java Enterprise Edition
JDBC	Java DataBase Connectivity
JDK	Java Development Kit
JSP	Java Server Pages
LOPD	Ley Orgánica de Protección de Datos de Carácter Personal
RF	Requisito Funcional
RI	Requisito Interfaz
RS	Requisito de Sistema
RSG	Requisito de Seguridad
RU	Requisito de Usabilidad
SO	Sistema Operativo
UC3M	Universidad Carlos III Madrid
UML	Unified Modeling Language
W3C	Worl Wide Web Consortium

12.1.2. Definiciones

Bootstrap	Framework creado por Twitter para modificar estilos CSS
Datasource	Ajustes de una conexión a una base de datos
e-Learning	Proceso de formación online
Gunning Fog	
Index	Índice que indica la edad apta para comprender un texto
Hardware	Parte física de un sistema informático
	Clases Java que implementa una funcionalidad en el servidor
Servlet	
Software	Parte lógica de un sistema informático
Taglib	Librerías de etiquetas para optimizar el código JSP

12.2. Anexo II: Requisitos

12.2.1. Requisitos funcionales

Identificador: RF-001	
Nombre: Introducir texto	
Necesidad: Esencial	Prioridad: Alta
Descripción: El usuario dispondrá de un cuadro de texto para introducir el texto a analizar.	

Tabla 26. RF-001 Introducir texto

Identificador: RF-002	
Nombre: Identificar usuarios	
Necesidad: Esencia	Prioridad: Alta
Descripción: El usuario podrá acceder a la plataforma con sus credenciales	

Tabla 27. RF-002 Identificar usuario

Identificador: RF-003	
Nombre: Registrar nuevos usuarios	
Necesidad: Esencial	Prioridad: Alta
Descripción: El usuario no registrado podrá crear una cuenta para acceder a las características de la plataforma para usuarios registrados.	

Tabla 28. RF-003 Registrar nuevos usuarios

Identificador: RF-004	
Nombre: Obtener informe	
Necesidad: Esencial	Prioridad: Alta
Descripción: El usuario obtendrá un informe con los detalles del análisis del texto introducido	

Tabla 29. RF-004 Obtener informe

Identificador: RF-005	
Nombre: Recuperar textos	
Necesidad: Deseable	Prioridad: Baja
Descripción: El usuario registrado podrá acceder a los textos introducidos con anterioridad.	

Tabla 30. RF-005 Recuperar textos

Identificador: RF-006	
Nombre: Recuperar informes	
Necesidad: Deseable	Prioridad: Media
Descripción: El usuario registrado podrá acceder a los informes generados de textos anteriores.	

Tabla 31. RF-006 Recuperar informes

Identificador: RF-007	
Nombre: Borrar textos	
Necesidad: Deseable	Prioridad: Baja
Descripción: El usuario podrá eliminar los textos guardados en su cuenta.	

Tabla 32. RF-007 Borrar textos

Identificador: RF-008	
Nombre: Modificar textos	
Necesidad: Deseable	Prioridad: Baja
Descripción: El usuario podrá modificar los textos guardados en su cuenta.	

Tabla 33. RF-008 Modificar textos

Identificador: RF-009	
Nombre: Analizar texto	
Necesidad: Esencial	Prioridad: Alta
Descripción: Utilizar la fórmula del índice Gunning Fog para analizar el texto introducido.	

Tabla 34. RF-009 Analizar texto

Identificador: RF-010	
Nombre: Cerrar sesión	
Necesidad: Deseable	Prioridad: Baja
Descripción: El usuario podrá cerrar la sesión en la plataforma cuando desee salir	

Tabla 35. RF-010 Cerrar sesión

Identificador: RF-011	
Nombre: Eliminar cuenta	
Necesidad: Opcional	Prioridad: Baja
Descripción: El usuario podrá eliminar su cuenta de la plataforma web.	

Tabla 36. RF-011 Eliminar cuenta

Identificador: RF-012	
Nombre: Eliminar informe	
Necesidad: Opcional	Prioridad: Baja
Descripción: El usuario podrá borrar los informes de su cuenta.	

Tabla 37. RF-012 Eliminar informe

Identificador: RF-013	
Nombre: Contenido personalizado	
Necesidad: Esencial	Prioridad: Alta
Descripción: Los usuarios dados de alta en el sistema, cuando accedan verán su propio contenido	

Tabla 38. RF-013 Contenido personalizado

Identificador: RF-014	
Nombre: Sesión iniciada	
Necesidad: Deseable	Prioridad: Media
Descripción: El usuario podrá ver la identificación utilizada para la sesión iniciada.	

Tabla 39. RF-014 Sesión iniciada

Identificador: RF-015	
Nombre: Clasificar textos	
Necesidad: Opcional	Prioridad: Media
Descripción: El usuario podrá clasificar los textos según su materia	

Tabla 40. RF-015 Clasificar textos

12.2.2. Requisitos no funcionales

12.2.2.1. Requisitos de sistema

Identificador: RS-001	
Nombre: Compatibilidad web	
Necesidad: Esencial	Prioridad: Alta
Descripción: La plataforma deberá ser accesible desde diferentes navegadores. En este caso, Internet Explorer 7.0, Google Chrome y Mozilla Firefox 7.0.	

Tabla 41. RS-001 Compatibilidad web

Identificador: RS-002	
Nombre: Base de datos	
Necesidad: Esencial	Prioridad: Alta
Descripción: Se utilizará un gestor de base de datos MySQL	

Tabla 42. RS-002 Base de datos

Identificador: RS-003	
Nombre: Tecnologías de desarrollo	
Necesidad: Esencial	Prioridad: Alta
Descripción: Para el desarrollo de la herramienta se utilizará tecnología HTML5, CSS3, Bootstrap y Taglibs.	

Tabla 43. RS-003 Tecnología de desarrollo

Identificador: RS-004	
Nombre: Entorno de desarrollo	
Necesidad: Deseable	Prioridad: Media
Descripción: Se utilizará el entorno de desarrollo Eclipse Luna para la implementación del proyecto	

Tabla 44. RS-004 Entorno de desarrollo

12.2.2.2. Requisitos de seguridad

Identificador: RSG-001	
Nombre: Sesión usuarios	
Necesidad: Esencial	Prioridad: Alta
Descripción: Cuando los usuarios registrados inicien sesión se deberán cargar sus datos. Así solo los usuarios identificados podrán acceder a datos almacenados en la plataforma. Así mismo, la sesión será cerrada cuando el usuario salga del sistema guardando los datos modificados del mismo.	

Tabla 45. RSG-001 Sesión usuarios

Identificador: RSG-002	
Nombre: Contraseñas	
Necesidad: Deseable	Prioridad: Alta
Descripción: Las contraseñas almacenadas deberán ser cifradas.	

Tabla 46. RSG-002 Contraseñas

12.2.2.3. Requisitos de usabilidad

Identificador: RU-001	
Nombre: Feedback usuarios	
Necesidad: Esencial	Prioridad: Alta
Descripción: Cuando los usuarios realicen tareas en la plataforma, el sistema les informará de lo sucedido a través de mensajes.	

Tabla 47. RU-001 Feedback usuarios

Identificador: RU-002	
Nombre: Diferenciación notificaciones	
Necesidad: Deseable	Prioridad: Alta
Descripción: Los mensajes de apoyo e información a los usuarios deberán ser claros y concisos. Para ello, se utilizará el mismo tipo de mensaje, pero con un título identificativo. Además, se diferenciarán los mensajes de error, los de advertencias y los informativos cambiando su color.	

Tabla 48. RU-002 Diferenciación notificaciones

Identificador: RU-003	
Nombre: Mensajes confirmación	
Necesidad: Deseable	Prioridad: Media
Descripción: Cuando el usuario realice una acción que conlleve una modificación importante, como borrar o modificar un texto se deberá mostrar un mensaje de confirmación.	

Tabla 49. RU-003 Mensajes confirmación

Identificador: RU-004	
Nombre: Lenguaje común	
Necesidad: Deseable	Prioridad: Media
Descripción: En toda la plataforma se utilizará un lenguaje sencillo que ayude a que todos los usuarios lo entiendan, dentro de una formalidad mínima.	

Tabla 50. RU-004 Lenguaje común

Identificador: RU-005	
Nombre: Guiar usuarios	
Necesidad: Deseable	Prioridad: Media
Descripción: Se deberá intentar que los usuarios cometan errores que son fáciles de evitar proporcionando las herramientas necesarias, sobre todo en la introducción y borrado de textos.	

Tabla 51. RU-005 Guiar usuarios

12.2.2.4. Requisitos de interfaz

Identificador: RI-001	
Nombre: Formulario inscripción	
Necesidad: Esencial	Prioridad: Alta
Descripción: La plataforma deberá tener un formulario para el registro de nuevos usuarios con los campos básicos de inscripción (nombre, apellidos, fecha de nacimiento, correo electrónico y contraseña)	

Tabla 52. RI-001 Formulario inscripción.

Identificador: RI-002	
Nombre: Formulario identificación	
Necesidad: Esencial	Prioridad: Alta
Descripción: La plataforma deberá tener un formulario con un campo para el correo electrónico y otro para la contraseña para que el usuario pueda acceder.	

Tabla 53. RI-002 Formulario identificación

Identificador: RI-003	
Nombre: Formulario subir texto	
Necesidad: Deseable	Prioridad: Alta
Descripción: La interfaz de la plataforma deberá mostrar un campo para subir textos	

Tabla 54. RI-003 Formulario subir texto

Identificador: RI-004	
Nombre: Botón eliminar texto	
Necesidad: Deseable	Prioridad: Media
Descripción: La interfaz de la plataforma deberá tener un botón para que el usuario pueda eliminar los textos almacenados en la plataforma.	

Tabla 55. RI-004 Botón eliminar texto

Identificador: RI-005	
Nombre: Ventana informe	
Necesidad: Esencial	Prioridad: Alta
Descripción: La interfaz de la plataforma deberá mostrar un campo para subir textos	

Tabla 56. RI-005 Ventana informe

12.3. Anexo III: Casos de uso

El objetivo de este anexo es mostrar los diagramas de los casos de uso explicados en el apartado 4.3 Casos de uso.

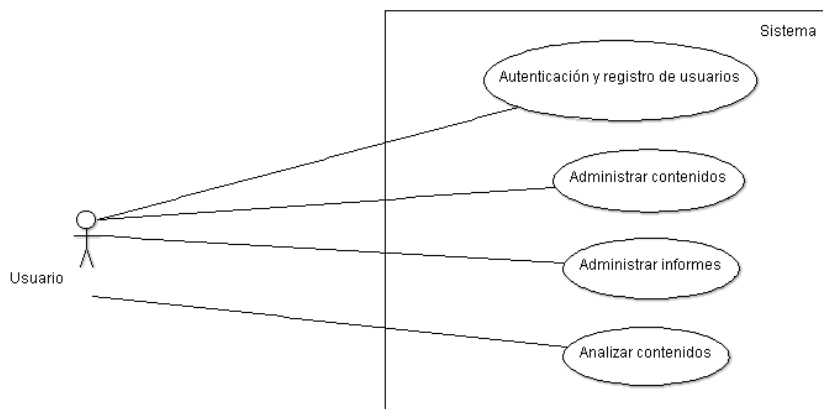


Ilustración 14. Caso de uso principal

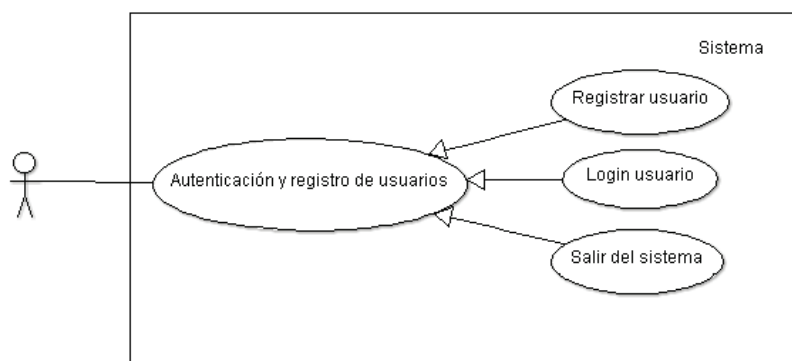


Ilustración 15. Caso de uso autenticación y registro de usuarios

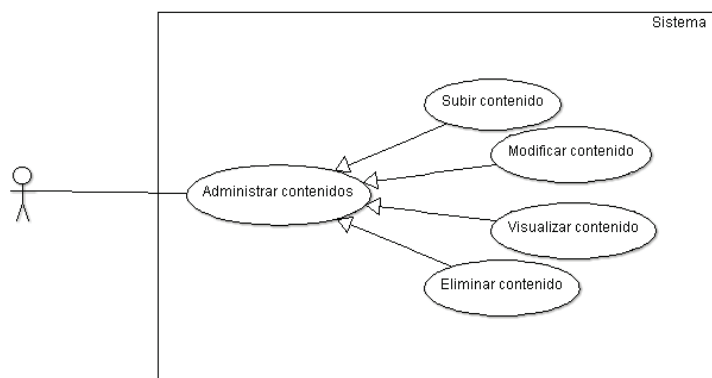


Ilustración 16. Caso de uso administrar contenidos

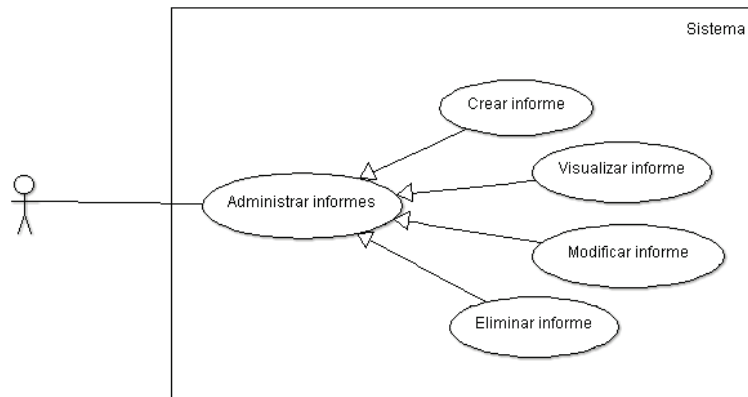


Ilustración 17. Caso de uso administrar informes

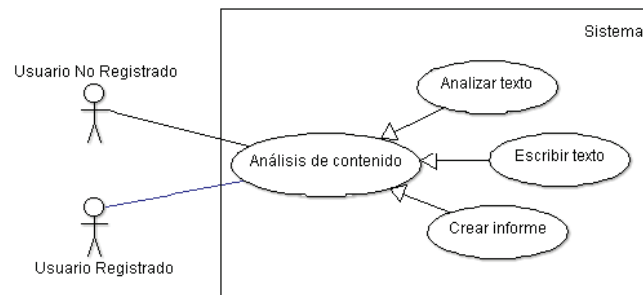


Ilustración 18. Caso de uso analizar contenido

12.4. Anexo IV: Matriz de trazabilidad

En este anexo se muestra la matriz de trazabilidad obtenida a partir de la relación entre los requisitos del sistema y los casos de uso establecidos.

	CU-01	CU-02	CU-03	CU-04	CU-05	CU-06	CU-07	CU-08	CU-09
RF-001				X	X				
RF-002		X							
RF-003	X								
RF-004								X	
RF-005					X	X	X		
RF-006								X	X
RF-007					X		X		X
RF-008				X	X		X	X	X
RF-009				X				X	
RF-010			X						
RF-011	X								
RF-012					X		X		
RF-013	X	X							
RF-014		X							
RF-015			X						

Tabla 57. Matriz de trazabilidad